# ARRAY SIGNAL PROCESSING USING HIGHER-ORDER AND FRACTIONAL LOWER-ORDER STATISTICS

by

Tsung-Hsien Liu

---

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(Electrical Engineering)

December 1998

# USC–SIPI REPORT #327

**Array Signal Processing Using Higher-Order and Fractional Lower-Order Statistics**
**by**

**Tsung-Hsien Liu**

**December 1998**

Signal and Image Processing Institute
**UNIVERSITY OF SOUTHERN CALIFORNIA**
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

# Dedication

This dissertation is dedicated to my family.

# Acknowledgements

I would like to express my gratitude to my advisor, Dr. Jerry M. Mendel, for his patient and valuable guidance through my Ph.D research. I appreciate his quick and sharp response to my research and this dissertation. He is one of the most influential people to my career. His academic and working attitude will continue to affect me in the future.

I would like to thank my qualifying exam committee members, Dr. Keith M. Chugg, Dr. Richard Leahy, Dr. Chrysostomos L. Nikias, and Dr. Chunming Wang. I would also like to thank Dr. Chugg and Dr. Wang for being on my defense committee.

I would like to thank the Center for Research on Applied Signal Processing for funding part of my research.

I would like to thank Dr. Mithat Dogan, Dr. Georgios Giannakis, Dr. Egemen Gonen, Dr. Daniel Rabideau, Dr. Gennedy Shamorodnitsky, Dr. Anathram Swami, and Dr. Panogiotis Tsakalides, for many useful technical discussions.

I would like to thank my officemates, Dr. Nilesh Karnik, Mr. Sunil Bharitkar, and Mr. Qilian Liang, for sharing happy memories in the office. Special thanks to Dr. Nilesh Karnik for discussing the LaTeX skills.

I would like to thank all the staff and computing services provided by the University of Southern California and Signal and Image Processing Institute.

Finally, I would like to thank my family for their support and encouragement.

# Contents

# List Of Tables

# List Of Figures

# Abstract

We apply higher-order statistics (HOS) and fractional lower-order statistics (FLOS) to different array signal processing scenarios.

In Chapter 2, we solve the two-angle (azimuth and elevation) direction of arrival (DOA) estimation problem using fourth-order cumulants. This work extends the virtual-ESPRIT algorithm (VESPA) [13] from one-angle (azimuth angle) to two angle DOA problems.

In Chapter 3, we discuss the sensitivity of VESPA. We are only interested in how VESPA is influenced by the model errors. Our analyses are useful when the finite-sample effects are small. We follow a conventional sensitivity procedure to perform our analyses, and derive sensitivity formulas for all the model parameters and output quantities.

In Chapter 4, we develop gradient-based target tracking using cumulants. The resulting subspace tracking algorithm has complexity of $\mathcal{O}(M^2P)$, where $M$ is the number of array elements and $P$ is the number of signals. We combine the resulting subspace tracking algorithm with VESPA to track moving targets.

In Chapter 5, we develop a cumulant-based preprocessing method that can be used with any data-matrix based algorithm to achieve rank and target tracking. We demonstrate the use of preprocessing with Rabideau's rank tracking algorithm [65] for both rank and target tracking. The resulting tracking algorithm has computational complexity of $\mathcal{O}(MP)$.

In Chapter 6, we consider the scenario where the additive noise is alpha stable. We construct several classes of fractional lower-order moment (FLOM) based matrices that can be used with MUSIC to extract DOAs. We assume that the signals are circular. From the simulation results, we conclude that the FLOM $p$ should be selected close to unity to yield best performance.

In Chapter 7, we construct a class of FLOM-based matrices that can be used with ESPRIT to extract DOAs. The scenario in this chapter is very similar to that in Chapter 6, except that there are two identical subarrays available. Analyses and simulations reveal that the FLOM-based ESPRIT shares similar properties with FLOM-based MUSIC.

In Chapter 8, we present our future research.

# Chapter 1

# Introduction

Array signal processing is a large collection of techniques that uses a group of spatially separated array sensors to extract information about signals that are in an environment. Because of the complicated physical components and various phenomena in the environment, array signal processing usually assumes a model for the environment. Each parameter in the assumed model represents a piece of information. The parameters of interest, depending on the applications and the models that are being used, include directions of arrival (DOAs), steering vectors, and the original transmitted signals. Accordingly, array signal processing detects or estimates the desired parameters from the observed measurements at array outputs.

In underwater acoustics, array sensors are called hydrophones, and signals are sound waves that travel through sea water. Typical applications of hydrophone arrays include source detection and localization, beamforming, array shape estimation, signal classification, and synthetic aperture sonar (SAS). Multipath effects are considered when the hydrophones are deployed in shadow waters. For more information about hydrophone array signal processing, see [89] and the references therein.

In mobile cellular communications, array sensors (at the mobile base stations) are array antennas, and the signals waves are electromagnetic (EM) waves that travel through free space. Typical applications include co-channel interference suppression and fading channel equalization. In code-division multiple-access (CDMA) mobile cellular systems, the use of array antennas also increases the capacity of the entire system [62, 60]. Because of fast-fading channels, adaptive signal processing is very important in these systems. More information and references about mobile communications using array antennas can be found in [21, 22].

In hand-free car phone or video conferencing systems, array sensors are microphones, and signals are human sounds that travel through free space. Typical problems include noise reduction, beamforming, and echo cancellation. Because array microphones are deployed in a small secluded compartment, multipath fading is more severe than in the preceding two scenarios. Signals are always modeled as wideband signals, due to the lack of modulation carriers, which appear in communication systems. Near-field effects are also considered due to the short distances between signals and microphones. All these properties lead to heavier computational burden than that required for narrowband far-field situations, because sophisticated array signal processing is required to utilize both spatial and temporal information carried by the array outputs. References for and adaptive techniques about hand-free phone systems can be found in [42].

The three scenarios (underwater acoustics, mobile communications, and hand-free phones) in the preceding paragraphs are just simple illustrations of array signal processing. There are many other applications of array signal processing. Books on array signal processing are [34, 36, 38, 42, 61], and review papers on the techniques used in array signal processing are [21, 22, 41, 89, 90].

The purposes of this dissertation are twofold. First, we extend the applications of array signal processing that use higher-order statistics (HOS), because HOS has been shown to possess some advantages over second-order statistics. Our extensions are focused on sensitivity analyses and adaptive implementations of some HOS-based algorithms. Second, we study the application of fractional lower-order moments (FLOMs) and fractional lower-order statistics (FLOS) to array signal processing scenarios where stable noise are present, because stable processes have bounded FLOMs and FLOS.

The structure of this chapter is as follows. Sections 1.1 and 1.2 list the abbreviations and symbols, respectively, that are used throughout this dissertation. The array model is introduced in Section 1.3. Literature that directly motivates our work is reviewed in Section 1.4. Overview of this dissertation is presented in Section 1.5.

## 1.1 Abbreviations

| | |
|---|---|
| AIC | Akaike's information criterion |
| CDMA | code-division multiple-access |
| CRLB | Cramer-Rao lower bound |
| DF | direction finding |
| DOA | direction of arrival |
| EM | electromagnetic |
| ESPRIT | estimation of signal parameters via rotational invariance technique |
| EVD | eigenvalue decomposition |
| FLOM | fractional lower-order moment |
| FLOS | fractional lower-order statistics |
| GSNR | generalized signal-to-noise ratio |
| GSO | Gram-Schmidt Orthogonalization |
| HOS | higher-order statistics |
| *i.i.d.* | independent identically distributed |
| LHS | left-hand side |
| LMS | least mean square |
| LS | least square |
| MC | Monte-Carlo |
| MDL | minimum description length |
| ML | maximum likelihood |
| MRS | method of refelction of subspace |
| MSE | mean squared error |
| MUSIC | multiple signal classification |
| MVDR | minimum variance distortionless response |
| PM | Phase modulation |
| QAM | quadrature amplitude modulation |
| RHS | right-hand side |
| ROC-MUSIC | robust covariation-based MUSIC |
| $S\alpha S$ | symmetric alpha stable |

| | |
|---|---|
| SAS | synthetic aperture sonar |
| SINR | signal-to-interference-plus-noise ratio |
| SNR | signal-to-noise ratio |
| STD | standard deviation |
| SVD | singular value decomposition |
| TLS | total least square |
| ULA | uniform linear array |
| VESPA | virtual-ESPRIT |

## 1.2 Symbols

| | |
|---|---|
| $z, Z$ | a scalar |
| $\mathbf{z}$ | a vector (a boldfaced lower-case letter) |
| $\mathbf{Z}$ | a matrix (a boldfaced upper-case letter) |
| $Z_{km}$, $\Phi_1(k,m)$ | the $(k,m)$-th entry of matrices $\mathbf{Z}$ and $\Phi_1$, respectively |
| diag $\{ \eta_1, \eta_2, \cdots, \eta_M \}$ | a diagonal matrix whose main diagonal entries are $\eta_1, \eta_2, \cdots, \eta_M$ |
| $E\{\cdot\}$ | expectation operator |
| $\mathbf{I}$ | identity matrix |
| $j$ | $\sqrt{-1}$ |
| $M$ | number of sensors in an array |
| $N$ | number of snapshots available |
| $n_i(t)$ | additive noise component at $i$-th sensor |
| $\Re\{\cdot\}$ | real part operator |
| $r_i(t)$ | array output at the $i$-th sensor |
| $s_p(t)$ | the $p$-th signal measured at the reference point |
| $P$ | number of signals present |
| $\mathbf{r}_i$ | spatial location of $r_i$ |
| tr($\mathbf{Z}$) | trace of a square matrix $\mathbf{Z}$ |
| $\gamma_{s_p}$ | the fourth-order cumulant of $s_p(t)$ |
| $\boldsymbol{\kappa}_p$ | propagation vector of signal $s_p(t)$ |
| $\lambda$ | wavelength of the carrier |
| $\phi_p$ | azimuth angle of signal $s_p(t)$ |

| | |
|---|---|
| $\pi$ | ratio of the circumference of a circle to its diameter |
| $\theta_p$ | elevation angle of signal $s_p(t)$ |
| $\angle g$ | phase (radian) of complex-valued $g$ |
| $\|g\|$ | absolute value of complex-valued $g$ |
| $\|\|\cdot\|\|$ | Euclidean norm of a vector or matrix |
| $(\cdot)^*$ | Complex conjugate of a complex number, complex vector, or complex matrix |
| $(\cdot)^T$ | transpose of a vector or a matrix |
| $(\cdot)^H$ | complex conjugate transpose of a vector or matrix |
| $\widehat{(\cdot)}$ | estimate of a scalar, vector, or matrix |

## 1.3 Array Model

The model introduced in this section is specific for communication systems. Models for other scenarios of array signal processing can be obtained if minor modifications are made to the model introduced in this section.

We introduce the analytic representation for communication signals, because it is very useful for analyzing such signals. Let $\bar{s}(t)$ be a modulated signal, defined as

$$\bar{s}(t) \triangleq m(t)\cos(2\pi f_c t + \psi(t)) \tag{1.1}$$

where $m(t)$ and $\psi(t)$ are the message signals (depending on the modulations), and, $f_c$ is the carrier frequency. The analytic representation, $s(t)$, of $\bar{s}(t)$ is

$$s(t) \triangleq \bar{s}(t) + j\bar{s}_h(t) \tag{1.2}$$

where $\bar{s}_h(t)$ is the Hilbert transform of $\bar{s}(t)$. Note that $\bar{s}(t)$ in (1.1) is a real signal, whereas its analytic representation $s(t)$ in (1.2) is a complex signal. Equation (1.2) defines how to convert a signal to its analytic representation.

A complex signal $s(t)$ is said to be analytic if it can be expressed in the form described by (1.2). From the real part of an analytic signal, we obtain its original signal, i.e., $\Re\{s(t)\} = \bar{s}(t)$. Accordingly, we can make conversions between a communication signal and its analytic representation.

Consider a scenario where $P$ co-channel emitters (signals) impinge on an $M$-element arbitrary array. The word "co-channel" means that all of the $P$ signals have the same carrier frequency. The signal received at the $i$-th sensor is

$$r_i(t) = \sum_{k=1}^{P} g_i(\boldsymbol{\theta}_k)\, s_k(t - \tau_i(\boldsymbol{\theta}_k)) + n_i(t) \qquad i = 1, \cdots, M \tag{1.3}$$

where

| | |
|---|---|
| $\boldsymbol{\theta}_k$ | DOA of $s_k(t)$; |
| $g_i(\boldsymbol{\theta}_k)$ | sensor gain with respect to signal $s_k(t)$; |
| $s_k(t)$ | the $k$-th signal measured at the reference point and expressed in analytic form; |
| $\tau_i(\boldsymbol{\theta}_k)$ | time delay for $s_k(t)$ to travel from the reference point to sensor $r_i$; |
| $n_i(t)$ | additive noise at $r_i(t)$. |

Note that in (1.3) $\boldsymbol{\theta}_k \triangleq [\phi_k\ \theta_k]^T$ when both azimuth $\phi_k$ and elevation $\theta_k$ angles are considered, and that $\boldsymbol{\theta}_k \triangleq \phi_k$ when only azimuth angle $\phi_k$ is considered.

Typically, the sensor gain $g_i(\boldsymbol{\theta}_k)$ depends on the characteristics of sensor $r_i$, and, is sometimes a very complicated function of $\boldsymbol{\theta}_k$ as well as other factors. In this dissertation, however, we assume that $g_i(\boldsymbol{\theta}_k)$ is a constant, $g_i$; hence, we use $g_i$ instead of $g_i(\boldsymbol{\theta}_k)$. Sensors are said to be *omnidirectional* if their sensor gains are constants. Two sensors are said to be *identical* if their sensor gains are identical.

The additive noise, $n_i(t)$, accounts for the thermal noise, measurement error, and all other phenomena that cannot be modeled by our assumptions.

If the signals are *narrowband* signals (i.e., the bandwidth of $s_k(t)$, $\forall k$, is small compared to $f_c$), then $s_k(t - \tau_i(\boldsymbol{\theta}_k)) \approx e^{-j2\pi f_c \tau_i(\boldsymbol{\theta}_k)} s_k(t)$ and (1.3) can be expressed as

$$r_i(t) \approx \sum_{k=1}^{P} g_i\, e^{-j2\pi f_c \tau_i(\boldsymbol{\theta}_k)}\, s_k(t) + n_i(t) \qquad i = 1, \cdots, M \tag{1.4}$$

If the signals are located in the *far-field* (i.e., the signals are located far away in comparison with the aperture of the entire array), wavefronts of the signals become planar and $2\pi f_c \tau_i(\boldsymbol{\theta}_k) = \boldsymbol{\kappa}_k^H \mathbf{r}_i$; in this case, (1.4) can be written as

$$r_i(t) = \sum_{k=1}^{P} g_i \, e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_i} \, s_k(t) + n_i(t) \qquad i = 1, \cdots, M \tag{1.5}$$

where

| | |
|---|---|
| $\boldsymbol{\kappa}_k$ | propagation vector of $s_k(t)$; |
| $\mathbf{r}_i$ | spatial location of $r_i$. |

Note that, in (1.5), the DOAs of $s_k(t)$, $\forall k$, are implicitly in $\boldsymbol{\kappa}_k$. The dimensions of $\boldsymbol{\kappa}_k$ and $\mathbf{r}_i$ depend on how problems are formulated, and explicit expressions for $\boldsymbol{\kappa}_k$, $\forall k$, in terms of DOAs, will be stated in later chapters. The locations of sensors, $\mathbf{r}_i$, $\forall i$, are measured in wavelengths of the carrier, $\lambda$. Note that the product of frequency $f_c$ and wavelength $\lambda$ is equal to the propagation speed of the propagating wavefront.

Array signal processing usually deals with baseband signals, not the signals at antenna outputs [see (1.5)]. The baseband signal of $r_i(t)$ is obtained by setting $f_c = 0$ in $s_k(t)$, $\forall k$ in (1.5) and (1.1). The baseband signals are also called the complex envelope representations of the received signals. We use complex envelope representations for all the signals in the rest of this dissertation.

The matrix form for (1.5) is

$$\mathbf{r}(t) = \mathbf{A}\,\mathbf{s}(t) + \mathbf{n}(t) \tag{1.6}$$

where

| | |
|---|---|
| $\mathbf{r}_{M\times 1}(t) \triangleq [r_1(t), r_2(t), \cdots, r_M(t)]^T$ | measurement vector (snapshot); |
| $\mathbf{A}_{M\times P} \triangleq \{A_{ik}\} = \{g_i \, e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_i}\}$ | array manifold; |
| $\mathbf{s}_{P\times 1}(t) \triangleq [s_1(t), s_2(t), \cdots, s_P(t)]^T$ | signal vector; |
| $\mathbf{n}_{M\times 1}(t) \triangleq [n_1(t), n_2(t), \cdots, n_M(t)]^T$ | noise vector. |

Equation (1.6) is common to all our later chapters. It is a linear system with $P$ inputs, $\mathbf{s}(t)$, and $M$ outputs, $\mathbf{r}(t)$. More explicit expressions for $A_{ik}$, $\forall i$ and $k$, will be given for the specific problems that are defined in later chapters. Note that in

arriving at (1.5) and (1.6), we assumed that the signals are located at far-field and are narrowband compared to the carrier frequency.

In subspace-based array signal processing, the array manifold, $\mathbf{A}$, is assumed to be of full rank. The column span of $\mathbf{A}$ is called the signal subspace, denoted by $\mathcal{R}\{\mathbf{A}\}$; the subspace that is orthogonal to $\mathcal{R}\{\mathbf{A}\}$ is called the noise subspace.

## 1.4 Literature Review

In this section, we do not intend to review all the literature on array signal processing; instead, we concentrate on the literature that is directly related to our work.

DOA estimation, also called direction finding (DF), is one of the most important and typical problems in array signal processing. Spectral estimation methods [57, 89] were first applied to DOA estimation, but they failed to resolve closely spaced signals. Maximum-likelihood (ML) criterion was then applied to the scenario when the noise is Gaussian and the signals are deterministic, or when the noise is Gaussian, and, the signals are jointly Gaussian but are independent of the noise [58, 73, 74]. The ML-estimators achieve their corresponding Cramer-Rao lower bounds (CRLBs), but they require solving multidimensional optimization problems. Even at this high cost of computations, the ML-estimators do not guarantee global convergence.

Subspace-based methods then appeared and have been shown to significantly outperform conventional spectral estimation methods, and to reduce computational costs, as compared with ML estimators. Subspace-based methods require that the signals are non-coherent (the correlation matrix of signals is non-singular); otherwise, spatial smoothing [70] must be applied to the spatial correlation matrix to avoid rank degeneracy. Another limitation for all subspace methods is that the number of signals to be detected must not exceed the number of array sensors. Among the methods developed, multiple signal classification (MUSIC) [68] and estimation of signal parameters via rotational invariance techniques (ESPRIT) [67] are the two most popular ones.

MUSIC [68] computes the signal or noise subspace by performing singular value decomposition (SVD) or eigenvalue decomposition (EVD) on the spatial correlation matrix of the array outputs. Assuming that the steering vector is known, MUSIC

solves a one (when azimuth angle is considered) or two (when both azimuth and elevation angles are considered) dimensional optimization problem for the parameters of interest. MUSIC has the same large-sample properties as the stochastic ML-estimator when the signals are uncorrelated [73]. The signal and noise subspaces can also be obtained by performing SVD on the data matrix. Root-MUSIC [2] uses a uniform linear array (ULA) and finds the roots of a polynomial; it avoids the search procedure, and, therefore, reduces computation load. Other implementations of MUSIC include beamspace MUSIC [44] and beamspace root-MUSIC [103].

ESPRIT [67] is a computationally more efficient algorithm for DOA estimation than MUSIC. Although it requires two identical subarrays, it algebraically calculates the DOAs and does not need to search over the parameter set; ESPRIT trades low computational costs for higher hardware costs. Other ESPRIT variants include beamspace ESPRIT [95], resolution-enhanced ESPRIT [31], and procrustes rotation-based ESPRIT [102].

The above-mentioned subspace methods are all based on second-order statistics. Some papers also report that the application of HOS to subspace-based methods can suppress additive Gaussian noise. For more information about the applications of HOS, see [52, 53, 56]. The use of fourth-order statistics (cumulants) attracts more attention than the use of third-order cumulants, because in array signal processing, the third-order cumulants are zero for most applications. There are cumulant-based MUSIC [5, 63], and cumulant-based ESPRIT methods [6, 7, 13, 30]. All these cumulant-based methods assume that the signals are non-Gaussian.

In [12], cumulants were used to suppress Gaussian interferers and additive noise, and to blindly estimate the steering vector that is associated with the desired non-Gaussian signal. The blindly estimated steering vector was then used to calculate the weights for the minimum-variance distortionless response (MVDR) beamformer [3]. It was also shown that the cumulant-based MVDR beamformer maximizes the signal-to-interference-plus-noise (SINR) ratio of array outputs.

In [13], a cumulant-based ESPRIT algorithm, called the virtual-ESPRIT algorithm (VESPA), was proposed to estimate the azimuth angles of independent non-Gaussian signals. It was shown that a copy of the main array, which can be very costly in terms of the additional hardware, is not necessary; it is possible to estimate

all the covariance terms that are needed in order to implement ESPRIT using cumulants, as long as there are only two identical sensors in the main array. This gave rise to a sequence of papers about applying cumulants to array signal processing.

In [14], a cumulant-based method was proposed to suppress spatially colored (possibly non-Gaussian) noise, if an additional sensor, whose measurement noise component is independent of the noise components of the original array measurements is available.

In [28], it was shown that cumulant-based blind beamforming can recover statistically independent narrowband signals in the presence of coherent (perfectly correlated) multipath propagation.

In [29], extended-VESPA [29] was developed; it deals with the scenario where coherent signals are present. Extended-VESPA is capable of resolving more signals than the number of array sensors. Extended-VESPA requires a uniform linear subarray, but the rest of the array may be arbitrarily configured.

In [25], a beamspace VESPA was proposed. In beamspace processing, the array data is projected into a beamspace of smaller rank than the number of array sensors; then, the projected beamspace data is processed to obtain DOAs in the same way as if it was received from a pseudo-array of lower dimensions. Beamspace VEPSA requires less computations than those for the element-space VESPA in [13].

In [26], iterative VESPA was proposed to handle the case where the data length is short and some of the signals have very small higher-order statistics as compared to others. This phenomenon is very similar to the near-far phenomenon in mobile cellular systems. For each iteration, iterative VESPA computes only the DOA and the steering vector associated with the most dominant signal. The data is then projected onto the beamspace that is orthogonal to the newly computed steering vector to yield beamspace data. In the next iteration of iterative VESPA, the processing is on this beamspace data. The total number of iterations is equal to the number of signals.

In [27], cumulant-based joint estimation of polarization parameters and DOAs was proposed. It is known that signals that are closely spaced but have different polarizations can be separated or resolved by using polarized antennas. If the polarization of the receiving antenna is matched with that of the signal, the induced power at the antenna output is maximized; similarly, if the two polarizations do not

match, the induced power at the receiver is very small. Diversity in polarization can be used to improve performance.

The above-mentioned works assume that the additive noise is Gaussian and the desired signals are non-Gaussian. Because the fourth-order cumulants of Gaussian processes are zeros, the use of cumulants is strongly recommended to remove the additive Gaussian noise. In some applications, however, a Gaussian model does not fully describe the noise phenomenon.

A recent study [71] shows that stationary symmetric alpha stable ($S\alpha S$) processes are better models for *impulsive* noise than are Gaussian processes. One of the major properties of $S\alpha S$ processes is that they do not have finite covariance and HOS. The use of covariance and HOS makes no sense, since statistical analyses cannot evaluate unbounded (infinite) quantities to obtain reasonable conclusions. For properties and applications of $S\alpha S$ processes, see [37, 69, 71] and the references therein.

In [83], an ML-estimator of DOAs was proposed for the scenarios where deterministic signals are embedded in $S\alpha S$ noise. The CRLB and a closed-form expression of the pseudo ML-estimator were derived when the $S\alpha S$ noise is Cauchy and its samples satisfy certain assumptions. The ML-estimator still requires heavy computations and does not guarantee global convergence. It was shown that the Cauchy beamformer greatly outperforms the Gaussian beamformer in a wide variety of $S\alpha S$ noise environments, and performs comparably to the Gaussian beamformer when the additive noise is Gaussian.

In [88], the robust covariation-based MUSIC (ROC-MUSIC) for DOA estimation used covariations and assumed that signals and additive noise are jointly $S\alpha S$. In fact, the covariation of two jointly $S\alpha S$ random variables can be expressed as a function of FLOMs when the $\alpha$-parameter is between 1 and 2; therefore, ROC-MUSIC can also be viewed as a method based on FLOMs. It was shown that ROC-MUSIC outperforms correlation-based MUSIC over of a wide range of $S\alpha S$ noise environments.

Other scenarios of array signal processing in the presence of $S\alpha S$ have also been investigated, including: wideband localization [84], broadband beamforming [85], joint DOAs and Doppler estimation [86], and adaptive beamforming [87].

## 1.5 Overview

Because DOA estimation is one of the most important problems in array signal processing, this dissertation focuses on a broad range of DOA-related problems.

In Chapter 2, we solve the two-angle (azimuth and elevation) DOA estimation problem using fourth-order cumulants. This work extends the work of [13] from one-angle (azimuth angle) to two angle DOA problems. Whereas the 1-D VESPA [13] reduces hardware costs by 50%, our 2-D VESPA requires $\frac{1}{3}$ of the hardware costs, as compared with covariance-based ESPRIT algorithms. To accomplish this requires that there be three identical sensors, which must not be located on a line, and, that the displacement vectors linking the sensors must be less than half a wavelength. We apply our 2-D VESPA to planar and volumetric arrays.

In Chapter 3, we discuss the sensitivity of VESPA. Sensitivity studies establish how the estimates, obtained from an algorithm, deviate from their true values in the presence of model errors and finite samples. We are only interested in how VESPA is influence by the model errors, because the influence of finite-sample effects on VESPA has been reported in [101]. Our analyses are useful when the finite-sample effects are small, i.e., when long data snapshots are available and the signal-to-noise ratio (SNR) is high. We follow a conventional sensitivity procedure to perform our analyses, and derive sensitivity formulas for all the model parameters and output quantities. These sensitivity formulas are compared with simulation results so as to verify their accuracy. We also compare the robustness of VESPA and ESPRIT, when both model errors are present and finite samples are available, for a wide range of scenarios, and show that VESPA is less sensitive to such errors than ESPRIT.

In Chapter 4, we develop gradient-based target tracking using cumulants. The motivation of using cumulants is to suppress additive Gaussian noise and to save hardware costs. Unlike the previous problems, we assume that the targets are moving, which leads to a non-stationary scenario. Our work solves an optimization problem for the desired signal subspace, using a stochastic gradient criterion. The resulting subspace tracking algorithm has complexity of $\mathcal{O}(M^2P)$, where $M$ is the number of array elements and $P$ is the number of signals. Convergence of our adaptive algorithm is briefly discussed. We combine the resulting subspace tracking algorithm with VESPA to track moving targets, and the complete target tracking

algorithm is called least-mean-squared VESPA (LMS-VESPA). Simulation results are provided to demonstrate the tracking ability of our LMS-VESPA.

In Chapter 5, we develop a cumulant-based preprocessing method that can be used with any data-matrix based algorithm to achieve rank and target tracking. The major difference between this work and that of Chapter 4 is: we are now able to do both rank and target tracking, whereas Chapter 4 assumes the number of signals are known, so that only target tracking is performed. We demonstrate the use of preprocessing with Rabideau's rank tracking algorithm [65] for both rank and target tracking. The resulting tracking algorithm has computational complexity of $\mathcal{O}(MP)$, which is less than that needed for Chapter 4's adaptive VESPA. Simulations results are reported for rank and target tracking using our cumulant-based preprocessing and Rabideau's tracking algorithm.

In Chapter 6, we consider the scenario where the additive noise is $S\alpha S$. We construct several classes of FLOM-based matrices that can be used with MUSIC to extract DOAs. Second-order statistics and HOS cannot be used here, because $S\alpha S$ processes do not have bounded second-order statistics and HOS's. We assume that the signals are circular, which is uncommon to subspace methods but is still reasonable for some practical applications. We compare the similarities and differences between our work and that in [88]. From the simulation results, we conclude that the FLOM $p$ should be selected close to unity to yield best performance.

In Chapter 7, we propose a FLOM-based ESPRIT, that is similar to the FLOM-MUSIC in Chapter 6, to estimate the DOAs of independent circular signals embedded in additive alpha stable noise at array outputs. We provide simulations for several values of $p$, and demonstrate that when $p$ is close to unity, the resulting FLOM-ESPRIT gives the best performance. Although FLOM-ESPRIT requires higher hardware costs than FLOM-MUSIC, it requires less computations than FLOM-MUSIC. FLOM-MUSIC and FLOM-ESPRIT share many similar properties.

In Chapter 8, we present our future research.

# Chapter 2

# Azimuth and Elevation Direction Finding Using Arbitrary Array Geometries

One of the most important problems in array signal processing is to determine the DOAs of far-field narrowband signals from array outputs. ESPRIT [67] has been greatly exploited because of computational efficiency. The use of ESPRIT to estimate one angle (i.e., azimuth angle) requires two identical subarrays. In the two-angle DOA problem (i.e., azimuth and elevation angles), two 1-D ESPRITs have been applied independently to the arrays, followed by an alignment procedure to associate the parameter estimates. The array can be made up of either three identical subarrays [79, 91] without constraints on array geometries, or two uniform linear subarrays [99], or a rectangular array [46]. In some cases, however, array hardware costs and *regular* array geometry constraints make the direction finding problem difficult.

VESPA [13] solved the single angle DOA problem. It was shown that a copy of the main array, which can be very costly in terms of the additional hardware, is not necessary; it is possible to estimate all the covariance terms that are needed in order to implement ESPRIT (or TLS-ESPRIT) using cumulants, as long as there are only two identical sensors in the main array. VESPA, therefore, saves a lot of hardware, and is applicable to arbitrary arrays.

In this chapter, we extend the single angle DOA problem in [13] to the two-angle case. We consider an array with arbitrary geometry in space and show how VESPA can be applied to the two-angle DOA problem.

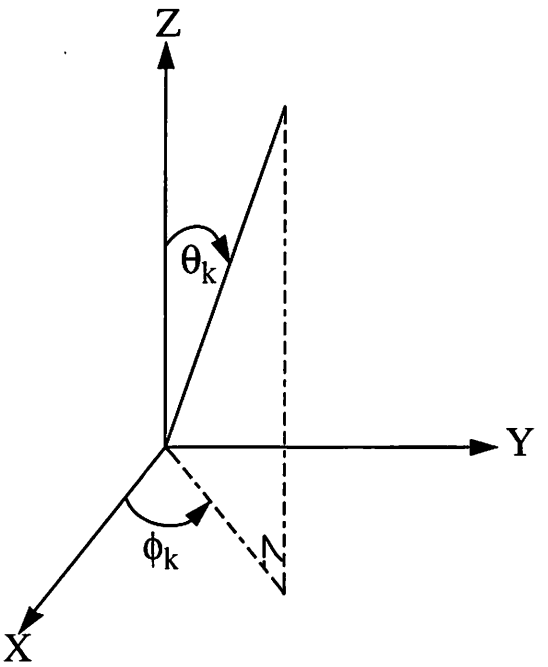



Figure 2.1: Azimuth $\phi_k$ and elevation $\theta_k$ angles characterize the direction of source $s_k(t)$.

In Section 2.1, we define the problem of interest. In Section 2.2, we describe how VESPA can be used to solve the two-angle DOA problem. Simulation results are presented in Section 2.3. Finally, we give our conclusions in Section 2.4.

## 2.1 Problem Formulation

We assume that the array sensors are already calibrated ($g_i = 1$, $\forall i$); From (1.5), the array outputs $r_i(t)$, $\forall i$, become

$$r_i(t) = \sum_{k=1}^{P} s_k(t)\, e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_i} + n_i(t), \qquad i = 1, 2, \cdots, M. \tag{2.1}$$

The propagation vector $\boldsymbol{\kappa}_k$ contains DOA information (see Fig. 2.1) and is defined as [4, 54]

$$\boldsymbol{\kappa}_k = 2\pi \left[\sin\theta_k \cos\phi_k,\ \sin\theta_k \sin\phi_k,\ \cos\theta_k\right]^T \tag{2.2}$$

where azimuth angle $\phi_k \in [-180°, 180°]$ and elevation angle $\theta_k \in [0, 90°]$; so DOA pair, $(\theta_k, \phi_k)$, characterizes the DOA of source $s_k(t)$. Signals $s_k(t)$, $\forall k$, are independent and non-Gaussian. We consider volume arrays in this chapter, and the sensor locations, $\mathbf{r}_i$, $\forall i$, are $3 \times 1$ vectors, and they are measured in wavelengths.

Similar to (1.6), we have the same matrix form for $\mathbf{r}(t)$,

$$\mathbf{r}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \tag{2.3}$$

but, the $\boldsymbol{\kappa}_k$, which appears in $A_{ik} \triangleq e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_i}$, is given by (2.2). The additive noise, $\mathbf{n}(t)$, is colored Gaussian with zero-mean and correlation matrix $\Sigma$.

Measurements are collected by sampling $\mathbf{r}(t)$ at N distinct times $t_n$, $n = 1, 2, \cdots, N$. The number of signals, $P$, is known in advance. The problem of interest is to estimate the DOA pairs $(\theta_k, \phi_k)$ for $s_k(t)$, $k = 1, 2, \cdots, P$, from measurements $\mathbf{r}(t_1), \mathbf{r}(t_2), \cdots, \mathbf{r}(t_N)$.

## 2.2 Cumulant-based ESPRIT Solution

Our cumulant-based solution consists of three steps:

1. construct three cumulant matrices, from which we estimate the "phase delay" matrices;
2. align the elements of the phase delay matrices;
3. determine DOAs from the phase delay matrices and the known locations of the guiding sensors.

### 2.2.1 Two 1-D VESPAs

To begin, we determine the fourth-order cumulant of the complex measurements $r_1(t)$, $r_2(t)$, $r_u(t)$, and $r_v(t)$, denoted $cum(r_1(t), r_2^*(t), r_u(t), r_v^*(t))$, which suppresses

the Gaussian components in $r_1(t)$, $r_2(t)$, $r_u(t)$, and $r_v(t)$. From (2.1) and cumulant properties in [52, 53], we have

$$\begin{aligned}
& cum(r_1(t), r_2^*(t), r_u(t), r_v^*(t)) \\
= \; & cum(\sum_{k=1}^{P} e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_1} s_k(t), (\sum_{l=1}^{P} e^{-j\boldsymbol{\kappa}_l^H \mathbf{r}_2} s_l(t))^*, \\
& \qquad \sum_{m=1}^{P} e^{-j\boldsymbol{\kappa}_m^H \mathbf{r}_u} s_m(t), (\sum_{n=1}^{P} e^{-j\boldsymbol{\kappa}_n^H \mathbf{r}_v} s_n(t))^*) \\
= \; & \sum_{k=1}^{P}\sum_{l=1}^{P}\sum_{m=1}^{P}\sum_{n=1}^{P} e^{-j\boldsymbol{\kappa}_k^H \mathbf{r}_1} e^{-j\boldsymbol{\kappa}_l^H \mathbf{r}_2} e^{-j\boldsymbol{\kappa}_m^H \mathbf{r}_u} e^{-j\boldsymbol{\kappa}_n^H \mathbf{r}_v} cum(s_k(t), s_l^*(t), s_m(t), s_n^*(t)) \\
= \; & \sum_{k=1}^{P} \gamma_{4,s_k} e^{j\boldsymbol{\kappa}_k^H(-\mathbf{r}_1+\mathbf{r}_2-\mathbf{r}_u+\mathbf{r}_v)} \qquad (2.4)
\end{aligned}$$

which follows from the independence of the source signals, i.e.,

$$cum(s_k(t), s_l^*(t), s_m(t), s_n^*(t)) = \begin{cases} \gamma_{4,s_k}, & \text{if } k = l = m = n \\ 0, & \text{otherwise} \end{cases} \qquad (2.5)$$

Three cumulant matrices, $\mathbf{C}_0$, $\mathbf{C}_1$, and $\mathbf{C}_2$, are then defined as follows:

$$\mathbf{C}_0 \triangleq cum(r_1(t), r_1^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}\mathbf{D}\mathbf{A}^H \qquad (2.6a)$$

$$\mathbf{C}_1 \triangleq cum(r_1(t), r_2^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}\boldsymbol{\Phi}_1\mathbf{D}\mathbf{A}^H \qquad (2.6b)$$

$$\mathbf{C}_2 \triangleq cum(r_1(t), r_3^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}\boldsymbol{\Phi}_2\mathbf{D}\mathbf{A}^H \qquad (2.6c)$$

where we have used (2.4), and,

$$\mathbf{D} \triangleq diag\{\gamma_{4,s_1}, \gamma_{4,s_2}, \cdots, \gamma_{4,s_P}\} \qquad (2.7)$$

$$\boldsymbol{\Phi}_1 \triangleq diag\{e^{j\boldsymbol{\kappa}_1^H(\mathbf{r}_2-\mathbf{r}_1)}, e^{j\boldsymbol{\kappa}_2^H(\mathbf{r}_2-\mathbf{r}_1)}, \cdots, e^{j\boldsymbol{\kappa}_P^H(\mathbf{r}_2-\mathbf{r}_1)}\} \qquad (2.8)$$

$$\boldsymbol{\Phi}_2 \triangleq diag\{e^{j\boldsymbol{\kappa}_1^H(\mathbf{r}_3-\mathbf{r}_1)}, e^{j\boldsymbol{\kappa}_2^H(\mathbf{r}_3-\mathbf{r}_1)}, \cdots, e^{j\boldsymbol{\kappa}_P^H(\mathbf{r}_3-\mathbf{r}_1)}\} \qquad (2.9)$$

We refer to the first two elements of $\mathbf{C}_0$, $\mathbf{C}_1$, and $\mathbf{C}_2$ as *guiding sensors*; more specifically, $r_1$ and $r_1$, $r_1$ and $r_2$, and, $r_1$ and $r_3$ are the three guiding sensor pairs for constructing $\mathbf{C}_0$, $\mathbf{C}_1$, and $\mathbf{C}_2$, respectively.

The steering matrix $\mathbf{A}$ is assumed to be of full rank; $\mathbf{D}$ is also of full rank, provided the sources have nonzero fourth-order cumulants. With these rank conditions, (2.6) satisfy all the requirements of the ESPRIT algorithm.

VESPA is now applied to $\mathbf{C}_0$, $\mathbf{C}_1$, and $\mathbf{C}_2$ to compute DOA information [13]. From (2.6), we define a matrix, $\mathbf{C}$, as

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{C}_0 \\ \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\Phi_1 \\ \mathbf{A}\Phi_2 \end{bmatrix} \mathbf{D}\mathbf{A}^H \tag{2.10}$$

The signal subspace $\mathbf{E}_s$ of the cumulant matrix $\mathbf{C}$ can be expressed as

$$\mathbf{E}_s \triangleq \begin{bmatrix} \mathbf{E}_0 \\ \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\Phi_1 \\ \mathbf{A}\Phi_2 \end{bmatrix} \mathbf{T} \tag{2.11}$$

where $\mathbf{T}$ is an invertible matrix. VESPA computes $\Psi_1 \triangleq \mathbf{T}^{-1}\Phi_1\mathbf{T}$ and $\Psi_2 \triangleq \mathbf{T}^{-1}\Phi_2\mathbf{T}$ by solving for

$$\mathbf{E}_1 = \mathbf{E}_0\Psi_1 \quad \text{and} \quad \mathbf{E}_2 = \mathbf{E}_0\Psi_2 \tag{2.12}$$

The EVD of $\Psi_1$ gives eigenvectors contained in $\mathbf{T}$, and eigenvalues along the diagonal of $\Phi_1$; the same decomposition of $\Psi_2$ gives $\mathbf{T}$ and $\Phi_2$. Since VESPA computes $\Phi_1$ and $\Phi_2$ independently, we refer to it as two 1-D VESPAs. However, because of the arbitrary ordering of the eigenvectors, the matrix $\mathbf{T}$ coming from the decomposition of $\Psi_1$ is, in general, not equal to the matrix $\mathbf{T}$ obtained from the decomposition of $\Psi_2$. We denote these two computed $\mathbf{T}$ matrices by $\mathbf{T}_1$ and $\mathbf{T}_2$, respectively. To correctly determine both the azimuth and elevation angles corresponding to the same source, we have to align the elements in matrices $\Phi_1$ and $\Phi_2$.

### 2.2.2 Alignment of Phase Delay Matrices

From (2.2), (2.8), and (2.9), we know that the DOA pair $(\theta_k, \phi_k)$ for source $s_k(t)$ is contained in the pair $(\Phi_1(k,k), \Phi_2(k,k)), k = 1, \cdots, P$. Unfortunately, independent

EVDs of $\Psi_1$ and $\Psi_2$ lead to arbitrary orderings of the elements of $\Phi_1(k,k)$ and $\Phi_2(k,k)$, $\forall k$. Fortunately, eigenvalues $\Phi_1(k,k)$ and $\Phi_2(k,k)$, $\forall k$, correspond to the same eigenvector [see (2.8) and (2.9)], which can then be used to restore the correct parameter pairs. Alignment of $\Phi_1(k,k)$ and $\Phi_2(k,k)$, $\forall k$, is therefore equivalent to alignment of the eigenvectors of $\Psi_1$ and $\Psi_2$.

Let the columns of $\mathbf{T}_1$ and $\mathbf{T}_2$ be normalized eigenvectors of $\Psi_1$ and $\Psi_2$, respectively. Because of the arbitrary ordering in the eigenvectors, permuting columns of $\mathbf{T}_1$ generates $\mathbf{T}_2$, and vice versa, i.e., $\mathbf{T}_1 = \mathbf{T}_2\,\mathbf{G}$, where $\mathbf{G}$ is a permutation matrix composed of a single unity along every row or column and zeros elsewhere. Challa and Shamsunder [6] computed

$$\mathbf{G} = \mathbf{T}_2^{-1}\,\mathbf{T}_1 \tag{2.13}$$

to learn the alignment information: wherever there is a unity at the $(i,j)$-th of $\mathbf{G}$, $\Phi_1(i,i)$ is aligned with $\Phi_2(j,j)$.

Our alignment procedure is based on projecting columns of $\mathbf{T}_1$, i.e., eigenvectors of $\Psi_1$, onto eigenspaces of $\Psi_2$, where eigenvectors are columns of $\mathbf{T}_2$. Let $\mathbf{t}_1$ be the first column of $\mathbf{T}_1$; then each entry in the complex projection row vector $\mathbf{t}_1^H\mathbf{T}_2$ represents the projection magnitude of $\mathbf{t}_1$ onto each eigenspace of $\Psi_2$. It is known that only one column $\mathbf{t}_2$, the eigenvector to be aligned with $\mathbf{t}_1$, of $\mathbf{T}_2$ is the same as $\mathbf{t}_1$. Projection magnitude of $\mathbf{t}_1$ into the eigenspace spanned by $\mathbf{t}_2$ is unity; projections of $\mathbf{t}_1$ into other eigenspaces will have their magnitudes less than unity since we are in the complex domain. In the presence of noise, we pick $\mathbf{t}_2$ such that the magnitude of $\mathbf{t}_1^H\mathbf{t}_2$ is closest to unity. Next, we form the eigenvalue pair $(\Phi_1(1,1), \Phi_2(1,1))$ where their associated eigenvectors are $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively, and remove the aligned eigenvectors $\mathbf{t}_1$ and $\mathbf{t}_2$ from $\mathbf{T}_1$ and $\mathbf{T}_2$, respectively. We repeat the same alignment procedure until all eigenvectors are aligned, and obtain the phase pairs $(\Phi_1(k,k), \Phi_2(k,k)), k = 1, \cdots, P$. This projection-based alignment procedure does not suffer from numerical instability caused by the matrix inversion in (2.13).

### 2.2.3 Determining DOAs From Phase Delay Matrices

So far, we have not made any assumption about the geometry of our array. Recall that $r_1$ is located at the reference point, which is also the origin of our coordinate system. We assume below that the guiding sensors $r_1$, $r_2$, and $r_3$ are located on the $x$-$y$ plane. Derivations similar to the one given below can be performed for the guiding sensors on the $y$-$z$ plane, $x$-$z$ plane, or even on a slanted plane. Recall the geometrical property that three points determine a plane if they are not on the same line or at the same point. Let $\Delta_1$ and $\Delta_2$ be known vectors connecting sensors $r_1$ and $r_2$, and $r_1$ and $r_3$, respectively; they can be decomposed into their $x$- and $y$-components, i.e.,

$$\Delta_1 \triangleq \mathbf{r}_2 - \mathbf{r}_1 = \delta_{1x}\,\mathbf{e}_x + \delta_{1y}\,\mathbf{e}_y \quad \text{and} \quad \Delta_2 \triangleq \mathbf{r}_3 - \mathbf{r}_1 = \delta_{2x}\,\mathbf{e}_x + \delta_{2y}\,\mathbf{e}_y \tag{2.14}$$

where $\mathbf{e}_x$ and $\mathbf{e}_y$ are unit vectors along the $x$- and $y$-axes, respectively. Substituting (2.2) and (2.14) into the phases of $(\Phi_1(k,k), \Phi_2(k,k))$, given in (2.8) and (2.9), we have

$$\begin{bmatrix} \angle\Phi_1(k,k) \\ \angle\Phi_2(k,k) \end{bmatrix} = 2\pi \begin{bmatrix} \delta_{1x} & \delta_{1y} \\ \delta_{2x} & \delta_{2y} \end{bmatrix} \begin{bmatrix} \sin\theta_k \cos\phi_k \\ \sin\theta_k \sin\phi_k \end{bmatrix}, \quad k = 1, \cdots, P \tag{2.15}$$

where $\angle\Phi_1(k,k)$ and $\angle\Phi_2(k,k)$ denote the phases of $\Phi_1(k,k)$ and $\Phi_2(k,k)$, respectively. Let

$$p_k \triangleq \sin\theta_k \cos\phi_k \quad \text{and} \quad q_k \triangleq \sin\theta_k \sin\phi_k \tag{2.16}$$

With known $\delta$'s, we can solve for $(p_k, q_k)$ from (2.15). The final step is to determine $(\theta_k, \phi_k)$ from $(p_k, q_k)$, i.e.,

$$\theta_k = \arcsin(\sqrt{p_k^2 + q_k^2}) \quad \text{and} \quad \phi_k = \arctan(q_k/p_k), \quad k = 1, \cdots, P. \tag{2.17}$$

The location matrix of $\delta$'s on the right-hand side (RHS) of (2.15) is guaranteed invertible by our prior assumption that guiding sensors are on the same plane. Furthermore, to uniquely determine $(p_k, q_k)$ from (2.15), we need

$$|\angle\Phi_1(k,k)| \leq \pi \quad \text{and} \quad |\angle\Phi_2(k,k)| \leq \pi. \tag{2.18}$$

Using the Schwarz inequality and $|\sin\theta_k| \leq 1$, it follows from (2.15) that

$$|\angle\Phi_1(k,k)| \leq 2\pi\sqrt{\delta_{1x}^2 + \delta_{1y}^2} \quad \text{and} \quad |\angle\Phi_2(k,k)| \leq 2\pi\sqrt{\delta_{2x}^2 + \delta_{2y}^2} \tag{2.19}$$

A *sufficient* condition, therefore, to avoid estimation ambiguity is

$$\sqrt{\delta_{1x}^2 + \delta_{1y}^2} < \tfrac{1}{2} \quad \text{and} \quad \sqrt{\delta_{2x}^2 + \delta_{2y}^2} < \tfrac{1}{2} \tag{2.20}$$

Note that the constraints in (2.20) are only on the guiding sensors, i.e., the length of the displacement vectors joining the guiding sensors must be less than half wavelength of the carrier; non-guiding sensors are unconstrained in terms of array geometry. These constraints are rather loose and provide us with lots of freedom in designing array shapes.

In practice, when applying (2.17) to solve for $\theta_k$ from $(p_k, q_k)$, $\sqrt{p_k^2 + q_k^2}$ can be greater than unity due to imperfect estimation of $(p_k, q_k)$. This, of course, causes the calculation of $\arcsin(\cdot)$ to fail. We call this situation *algorithm failure*, and shall examine it further in Section 2.3.2.

### 2.2.4 Summary of Least-Squares VESPA

The following steps, which come from a straightforward substitution of estimated values into (2.6)-(2.17), are termed a *least-squares implementation* of VESPA, or LS-VESPA:

1. Compute estimates of $\mathbf{C_0}$, $\mathbf{C_1}$, and $\mathbf{C_2}$, defined in (2.6).
2. Perform SVD on $\mathbf{C}$ to obtain $\mathbf{E}_s$ in (2.11).
3. Compute $\Psi_1$ and $\Psi_2$ using (2.12).

4. Eigendecompose $\Psi_1$ ($\Psi_2$) to obtain $\Phi_1$ and $\mathbf{T}_1$ ($\Phi_2$ and $\mathbf{T}_2$).

5. Align $(\Phi_1(k,k), \Phi_2(k,k))$, $k = 1, \cdots, P$.

6. Compute $(\theta_k, \phi_k), k = 1, \cdots, P$, using (2.15)-(2.17).

## 2.3 Simulations

### 2.3.1 LS-VESPA Applied to Various Array Configurations

In our first experiment, we tested LS-VESPA using the two array configurations, listed in Table 2.1 and depicted in Fig. 2.2.

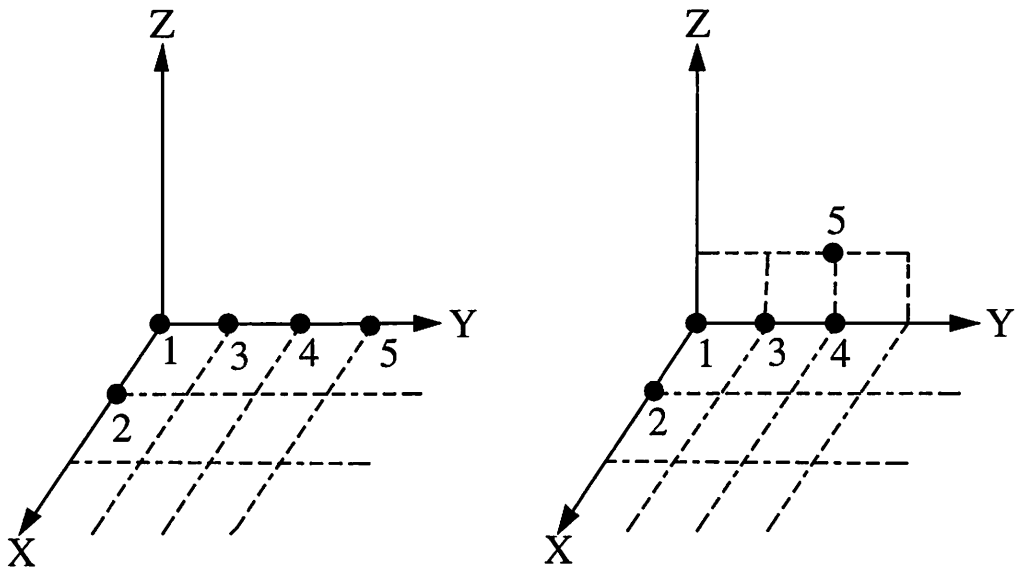
Table 2.1: Array configurations used in the experiments. These sensor positions are measured in wavelengths.

| Sensor No. | Configuration 1 | Configuration 2 |
|---|---|---|
| 1 | (0,0,0) | (0,0,0) |
| 2 | (1/2,0,0) | (1/2,0,0) |
| 3 | (0,1/2,0) | (0,1/2,0) |
| 4 | (0,1,0) | (0,1,0) |
| 5 | (0,3/2,0) | (0,1,1/2) |

There are three independent sources arriving from the directions listed in Table 2.2. Each signal is represented by $s_k(t) = s_k e^{j\psi_k(t)}$, where $s_k$ is the amplitude, and $\psi_k(t)$ is a sequence of independent identically distributed (*i.i.d.*) random variables that are uniform over [0,2$\pi$). Each measurement is contaminated by complex symmetrical Gaussian noise that is both spatially and temporally white and independent of the sources, i.e., the spatial noise covariance $\Sigma$ for $\mathbf{n}(t)$ is $\sigma^2\mathbf{I}$. The three sources have the same signal-to-noise ratios (SNR) with respect to the measurement

Table 2.2: DOAs for the three sources in the experiments.

| Source No. | $\phi_k$ | $\theta_k$ |
|---|---|---|
| $k = 1$ | 60° | 70° |
| $k = 2$ | 40° | 50° |
| $k = 3$ | 20° | 30° |




(a) Configuration 1 (b) Configuration 2

Figure 2.2: Two array configurations.

noise; 2000 snapshots were collected from the sensor outputs; and, 50 Monte Carlo runs were performed for the two different array configurations.

We chose the first three sensors as guiding sensors; they form a plane and are the same for both configurations. The displacement vectors joining the two guiding sensor pairs are along $x$- and $y$-axes, respectively, and their magnitudes are half wavelength. Configuration 1 is a planar array and Configuration 2 is a volume array.

We investigated each configuration for SNR = 0, 10, and 20 dB. Sample means and standard deviations (STDs) of the DOA estimates were computed for each experiment for performance evaluation, and are summarized in Tables 2.3 and 2.4. The simulations show that our algorithm works well for the two configurations. It is reasonable that sample STDs of the estimates increase as SNR decreases since noisy data degrades performance.

### 2.3.2 Failure Rate

When applying (2.17) to solve for $(\theta_k, \phi_k)$, our algorithm breaks down if $\sqrt{p_k^2 + q_k^2} > 1$, which, of course, is different from the phase unwrapping error for $\phi_k$, where

Table 2.3: Statistics of the estimates for Configuration 1.

| Angle | SNR = 0 dB | | SNR = 10 dB | | SNR = 20 dB | |
|---|---|---|---|---|---|---|
| | mean | STD | mean | STD | mean | STD |
| $\hat{\phi}_1$ | 59.9409° | 1.0718° | 59.9504° | 0.4668° | 60.0715° | 0.2889° |
| $\hat{\theta}_1$ | 70.2869° | 4.1680° | 70.1700° | 1.6032° | 70.1709° | 0.9961° |
| $\hat{\phi}_2$ | 40.3035° | 1.0558° | 40.0743° | 0.4233° | 39.9445° | 0.3472° |
| $\hat{\theta}_2$ | 50.1336° | 1.9738° | 50.1546° | 0.5546° | 49.9936° | 0.4721° |
| $\hat{\phi}_3$ | 19.9988° | 2.2703° | 19.8605° | 1.1618° | 19.9483° | 0.7373° |
| $\hat{\theta}_3$ | 30.0311° | 1.6619° | 29.8942° | 0.4683° | 30.0057° | 0.2670° |

estimates are available but are erroneous. Phase unwrapping error, which happens when $\phi_k$ is close to $-180°$ or $180°$, has been addressed by others, e.g., [94], for the 1-D DOA problem. Here we are only interested in the situation when estimates of $\theta_k$ are unavailable due to $\sqrt{p_k^2 + q_k^2} > 1$.

In this experiment, we considered Configuration 1 for the case of a single source: azimuth angle $\phi$ was fixed at 20° and elevation angle $\theta$ was varied from 70° to 90° in 1° increments. For each $(\theta, \phi)$, we conducted 50 trials and counted the number of times our algorithm failed. The average percentage failure rate, i.e., total number of failures divided by 50, and then multiplied by 100, is depicted in Fig. 2.3 for SNR = 0, 10, 20 dB.

Table 2.4: Statistics of the estimates for Configuration 2.

| Angle | SNR = 0 dB | | SNR = 10 dB | | SNR = 20 dB | |
|---|---|---|---|---|---|---|
| | mean | STD | mean | STD | mean | STD |
| $\hat{\phi}_1$ | 58.8286° | 1.3208° | 59.9483° | 0.5466° | 60.0147° | 0.3738° |
| $\hat{\theta}_1$ | 69.2298° | 5.9310° | 70.0660° | 1.9364° | 70.1261° | 1.7619° |
| $\hat{\phi}_2$ | 40.1743° | 6.6358° | 40.0373° | 0.5206° | 40.0077° | 0.4147° |
| $\hat{\theta}_2$ | 50.9651° | 8.6457° | 50.1851° | 0.7321° | 50.0628° | 0.4827° |
| $\hat{\phi}_3$ | 21.8310° | 4.7448° | 19.8633° | 1.3520° | 19.8960° | 1.2296° |
| $\hat{\theta}_3$ | 30.6252° | 3.0915° | 29.9375° | 0.4625° | 29.9871° | 0.3571° |

Observe that the closer $\theta$ is to 90°, the higher the failure rate is for all SNRs. This phenomenon can be explained from the parameterization of $(\theta, \phi)$ [see (2.16)]. The range for $(p_k, q_k)$ is the area inside the unit circle (see Fig. 2.4). Increasing

$\theta_k$ causes $(p_k, q_k)$ to be closer to the unit circle. Noise causes the estimate $(\hat{p}_k, \hat{q}_k)$ to deviate from its true value. The closer $(p_k, q_k)$ is to the unit circle, the more likely it is that $(\hat{p}_k, \hat{q}_k)$ will cross the unit circle, resulting in algorithm failure. Large amounts of noise cause $(\hat{p}_k, \hat{q}_k)$ to more easily move outside of the unit circle; so, at fixed $\theta$, our algorithm has a higher failure rate for low SNR (0 dB) than for high SNR (20 dB). Moreover, the failure rate of 50 % when $\theta$ is close to 90° implies that the *white noise* causes the estimate of $(\hat{p}_k, \hat{q}_k)$ to deviate in all directions.

Although we have presented numerical results for $\phi = 20°$, our conclusions are valid for all $\phi$.

## 2.4 Conclusions

We have demonstrated that a cumulant-based ESPRIT method can be used to estimate both the azimuth and elevation angles of independent non-Gaussian far-field sources. This extension of the VESPA algorithm to the two angle problem inherits the advantages of the original VESPA, including no constraints on the array geometry and a reduction of total hardware. Our algorithm can estimate azimuth and elevation angles of $M - 1$ sources using $M$ sensors. To avoid estimation ambiguity, the length of the guiding vectors linking the guiding sensors are required to be less than half the wavenumber of the carrier frequency. The simulations show that the algorithm works well for two tested array configurations. Furthermore, when the source direction is at low elevation (i.e., $\theta = 90°$), our algorithm is likely to fail. We presented a parametric model to explain this phenomenon, and verified it via simulations.

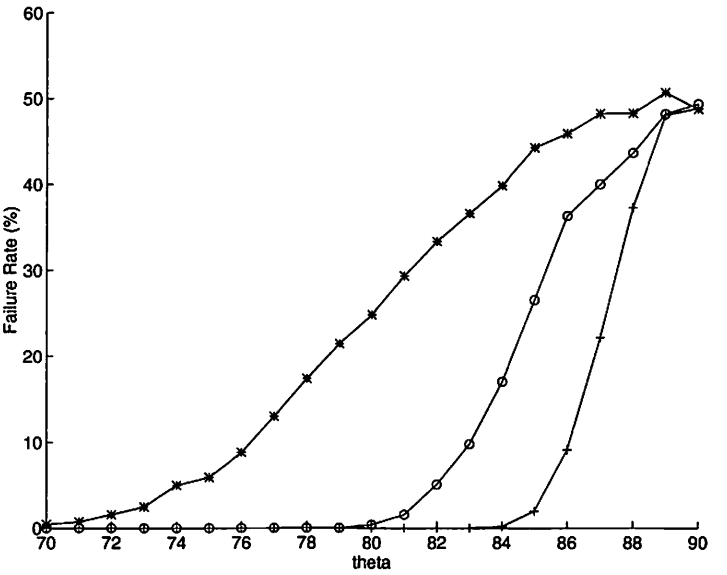



Figure 2.3: Failure rate versus $\theta$ (degrees) for different SNRs: 20 dB (+), 10 dB (o), and 0 dB (*).

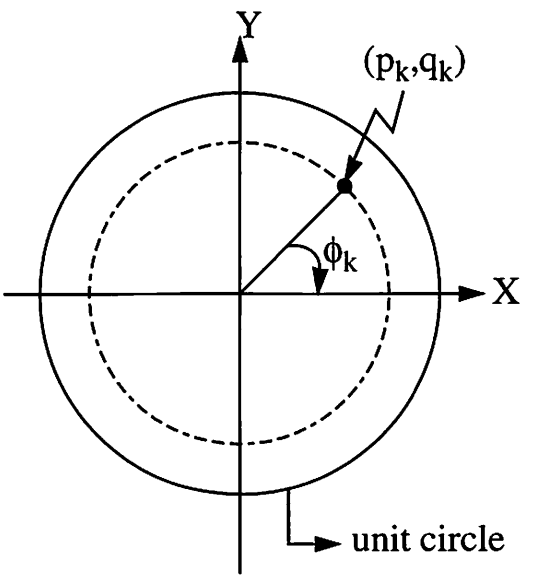



Figure 2.4: The black dot denotes the location of $(p_k, q_k)$, defined in (2.16). Magnitude of the vector $(p_k, q_k)$ is $\sin\theta_k$, and, phase of $(p_k, q_k)$ is $\phi_k$ measured relative to the $x$-axis; therefore, the range of $(p_k, q_k)$ is the region inside of the unit circle.

# Chapter 3

# Sensitivity of Virtual-ESPRIT in the Presence of Model Errors

Subspace algorithms for array signal processing start with computing a spatial correlation matrix. In practice, perfect knowledge of the spatial correlation matrix is unavailable, due to unavoidable noise and model parameter uncertainty. Several analyses [58, 66, 73], assuming perfect knowledge of model parameters, consider only the effect of finite samples due to unavoidable noise in the measurements. Some researchers [18, 32, 47, 78, 80], assuming perfect knowledge of spatial correlation matrix, consider only the effect of model parameter errors. Also, some analyses [45, 92] consider the combined effects of finite samples and model errors. It is clear that these two effects dominate the performance of all algorithms. In case that the SNR is high and measurement data is long enough, the effect of finite samples can be ignored and the effect of model errors dominate. On the other hand, if accurate model parameters are available through precise calibration, model errors can be ignored and the finite-sample effect dominates.

VESPA only requires two identical sensors, and is, therefore, expected to be robust to model errors or model perturbations. Yuen and Friedlander [101] compared the sensitivities of ESPRIT, Chiang and Nikias's [7] cumulant-based ESPRIT, and VESPA with respect to finite-sample effect. In this chapter, we study the sensitivity of VESPA with respect to model errors.

Section 3.1 introduces the procedure and the model used in our analysis, and briefly reviews the steps to implement VESPA. In Section 3.2, we derive closed-form formulas for the sensitivity. Approximated small-perturbation formulas are

given in Section 3.3; they yield simpler expressions of the exact sensitivity formulas. Computer simulations are presented in Section 3.4 to verify the accuracy of the theoretical analysis. Conclusions are given in Section 3.5.

## 3.1 Problem Description

It is known that a sensitivity analysis depends greatly on the assumed model. In this section, first the classical sensitivity analysis procedure is introduced; second, a specific definition of the array model is defined; third, a review of the detailed steps for a realization of VESPA is given; and, fourth, a perturbation model is introduced.

### 3.1.1 Conventional Sensitivity Analysis Procedure

A conventional sensitivity analysis procedure is comprised of the following steps:

**[CS1]** define an assumed model with parameters $\{g_i\}_{i=1}^{M}$, and output quantities of interest $\{\phi_p\}_{p=1}^{P}$;

**[CS2]** apply an input to the assumed model, and compute its outputs $\{\phi_p\}_{p=1}^{P}$;

**[CS3]** perturb a single model parameter, say $g_i$, such that $g_i$ becomes $\tilde{g}_i$, and keep the remaining parameters unchanged;

**[CS4]** apply the same input to this perturbed model and compute its perturbed outputs $\{\tilde{\phi}_p\}_{p=1}^{P}$; therefore, $g_i$ corresponds to $\{\phi_p\}_{p=1}^{P}$ and $\tilde{g}_i$ corresponds to $\{\tilde{\phi}_p\}_{p=1}^{P}$;

**[CS5]** compute the sensitivity of $\phi_p$ with respect to $g_i$, $S_{g_i}^{\phi_p}$, using the formula

$$S_{g_i}^{\phi_p} \triangleq \frac{\tilde{\phi}_p - \phi_p}{\phi_p}, \qquad p = 1, 2, \cdots, P; \tag{3.1}$$

**[CS6]** repeat **[CS3]** through **[CS5]** for each perturbed parameter $\tilde{g}_i, i = 1, 2, \cdots, M$, defined in **[CS1]**.

Note that (3.1) is valid for both large and small perturbations. The sensitivity is a measure of normalized deviation of the desired output caused by perturbations.

Note, also, that some people define $S_{g_i}^{\phi_p}$ as $\frac{\tilde{\phi}_p - \phi_p}{\phi_p} / \frac{\tilde{g}_i - g_i}{g_i}$. The division by $(\tilde{g}_i - g_i)/g_i$ is only a scaling of $S_{g_i}^{\phi_p}$ in (3.1).

To investigate the situation when multiple model parameters are perturbed, we can perturb the multiple parameters (step **[CS3]**) simultaneously, compute the perturbed quantity (step **[CS4]**), and then calculate $(\tilde{\phi}_p - \phi_p)/\phi_p$ on the RHS of (3.1). The symbol $S_{g_i}^{\phi_p}$ is inadequate for multiple perturbations, because not only $g_i$ but also all $\{g_i\}_{i=1}^M$ are perturbed. Throughout this chapter, therefore, we use $S_{g_i}^{\phi_p}$ when only $g_i$ is perturbed, and $(\tilde{\phi}_p - \phi_p)/\phi_p$ when multiple perturbations occur. Moreover, perturbation $\tilde{g}_i$ can also be random. For random variable $\tilde{g}_i$, we can compute the statistics of $\{S_{g_i}^{\phi_p}\}_{p=1}^P$. Finally, it is general and widely adopted in the literature [18, 32, 47, 78, 80] that random perturbations occur for multiple parameters; hence, the statistics of $\{(\tilde{\phi}_p - \phi_p)/\phi_p\}_{p=1}^P$ will be studied.

### 3.1.2 Nominal Data Model

Consider that $P$ far-field narrowband non-Gaussian independent sources $\{s_p(t)\}_{p=1}^P$ at directions $\{\phi_p\}_{p=1}^P$ impinge on a planar array of $M$ elements. The complex envelope representation for signal $r_i(t)$ received by the $i$-th sensor $r_i$ is modeled as

$$r_i(t) = \sum_{p=1}^{P} A_{ip}\, s_p(t) + n_i(t) \qquad i = 1, \cdots, M \tag{3.2}$$

where $A_{ip}$ is the sensor response of $r_i$ with respect to source $s_p(t)$, and $n_i(t)$ is the Gaussian measurement noise. The sensor response $A_{ip}$ is often modeled as

$$A_{ip} \triangleq g_i\, e^{-j2\pi[r_{ix}\sin\phi_p + r_{iy}\cos\phi_p]}, \qquad i = 1, \cdots, M \quad \text{and} \quad p = 1, \cdots, P \tag{3.3}$$

where $g_i$ is the sensor gain, and, $r_{ix}$ and $r_{iy}$ (real numbers) are the spatial locations of sensor $r_i$ along $x$- and $y$-axes, respectively. We use omnidirectional sensors in the following analyses, i.e., $\{g_i\}_{i=1}^M$ are constants. Recall that two omnidirectional sensors, $r_1$ and $r_2$, are said to be *identical* if $g_1 = g_2$. Note that in (3.2) and (3.3), we have assumed that only azimuth angle is considered and that a planar array is used. Substituting $\boldsymbol{\kappa}_p^H = [\sin\phi_p \;\; \cos\phi_p]$ and $\mathbf{r}_i = [r_{ix}\; r_{iy}]^T$ into (1.5), we have (3.2).

Without loss of generality, we make the following assumptions:

**[AS1]** Relative sensor gains with respect to $g_1$ are considered, i.e., $g_1 = 1$;

**[AS2]** Because of the special roles of guiding sensors in VESPA, sensors $r_1$ and $r_2$ are the guiding sensors in our analysis;

**[AS3]** The first sensor $r_1$ is chosen to be the reference sensor located at the origin of the coordinate system, i.e., $r_{1x} = r_{1y} = 0$;

**[AS4]** The $x$-axis is chosen to coincide with the displacement vector connecting guiding sensors $r_1$ and $r_2$; hence, $r_{2y} = 0$;

**[AS5]** The broadside direction ($\phi = 0°$) is defined to be the direction normal to the $x$-axis.

When **[AS1]** and **[AS3]** are substituted into (3.3), we find that

$$A_{1p} = 1, \qquad p = 1, 2, \cdots, P. \tag{3.4}$$

Collecting signals from array outputs, we have the following vector form of the received signal model:

$$\mathbf{r}(t) = \mathbf{A}\,\mathbf{s}(t) + \mathbf{n}(t) \tag{3.5}$$

which is exactly the same as (1.6), but $A_{ip}$ is given by (3.3).

It is clear, from (3.3) and our previous discussions, that the model parameters of interest during a sensitivity analysis are: sensor gains $\{g_i\}_{i=2}^{M}$, sensor positions along the $x$-axis $\{r_{ix}\}_{i=2}^{M}$, and sensor positions along the $y$-axis $\{r_{iy}\}_{i=2}^{M}$. The output quantities of interest can be DOAs, if the problem of interest is direction finding; or, the output quantities can be source signals, if the problem of interest is signal copy. Here, we consider DOA estimation only, which leads to the output quantities to be DOAs $\{\phi_p\}_{p=1}^{P}$. This completes **[CS1]** of the classical sensitivity procedure.

Note that in [13], it is assumed that: (1) the guiding sensors are already calibrated (or identical); and, (2) the displacement vector connecting guiding sensor pair $r_1$ and $r_2$ is known, and has length less than half-wavelength of the carrier, in order to avoid spatial aliasing. If these two assumptions hold, VESPA should accurately estimate the DOAs. The work of sensitivity analysis is to quantify the influence on the

output quantities (DOAs) of undesired perturbations of the model parameters, i.e., to determine $S_{g_i}^{\phi_p}$ for $\{g_i\}_{i=1}^{M}$ and $\{\phi_p\}_{p=1}^{P}$.

### 3.1.3 Virtual-ESPRIT [13, 29]

Assuming non-Gaussian independent signals $\{s_p(t)\}_{p=1}^{P}$ with fourth-order cumulants $\{\gamma_{4,s_p}\}_{p=1}^{P}$, two cumulant matrices, $\mathbf{C}_1$ and $\mathbf{C}_2$, are defined as follows:

$$\mathbf{C}_1 \triangleq cum(r_1(t), r_1^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}\Lambda\mathbf{A}^H \tag{3.6a}$$

$$\mathbf{C}_2 \triangleq cum(r_1(t), r_2^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}\Phi\Lambda\mathbf{A}^H \tag{3.6b}$$

where [29]

$$\begin{aligned}\Phi &\triangleq \mathrm{diag}\{\frac{A_{21}^*}{A_{11}^*}, \frac{A_{22}^*}{A_{12}^*}, \cdots, \frac{A_{2P}^*}{A_{1P}^*}\} \\ &= |g_2|\ \mathrm{diag}\{e^{j[2\pi(r_{2x}\sin\phi_1 + r_{2y}\cos\phi_1) - \angle g_2]}, e^{j[2\pi(r_{2x}\sin\phi_2 + r_{2y}\cos\phi_2) - \angle g_2]}, \\ &\qquad \cdots, e^{j[2\pi(r_{2x}\sin\phi_P + r_{2y}\cos\phi_P) - \angle g_2]}\}\end{aligned} \tag{3.7}$$

and

$$\begin{aligned}\Lambda &\triangleq \mathrm{diag}\{\gamma_{4,s_1}|A_{11}|^2, \gamma_{4,s_2}|A_{12}|^2, \cdots, \gamma_{4,s_P}|A_{1P}|^2\} \\ &= \mathrm{diag}\{\gamma_{4,s_1}, \gamma_{4,s_2} \cdots, \gamma_{4,s_P}\},\end{aligned} \tag{3.8}$$

since $A_{1p} = 1,\ \forall p$.

We refer to the first two elements of $\mathbf{C}_1$ and $\mathbf{C}_2$ as *guiding sensors*; more specifically, $r_1$ and $r_1$, and $r_1$ and $r_2$ are the guiding sensor pairs for constructing $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively. Although $r_{2y} = 0$ by our earlier assumption, $r_{2y}$ is not removed from (3.7), because $r_{2y}$ is to be replaced by $\tilde{r}_{2y}$ when perturbations are considered.

The steering matrix $\mathbf{A}$ is assumed to be of full rank. $\Lambda$ is also of full rank, provided the sources have nonzero fourth-order cumulants. With these rank conditions, (3.6)-(3.8) satisfy all the requirements of the ESPRIT algorithm. VESPA is now applied to $\mathbf{C}_1$ and $\mathbf{C}_2$ to compute DOA information [13]. The detailed steps of VESPA are [29]:

1. Estimate the $2M \times M$ cumulant matrix $\mathbf{C} \triangleq [\mathbf{C}_1^T\ \mathbf{C}_2^T]^T$ defined in (3.6).

2. Perform SVD on $\mathbf{C}$; retain those left singular vectors associated with the $P$ largest singular values in $\mathbf{E}_s$; partition $\mathbf{E}_s$ into two $M \times P$ matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, i.e., $\mathbf{E}_s \triangleq [\mathbf{E}_1^T \ \mathbf{E}_2^T]^T$. The column space of $\mathbf{E}_s$ is the so-called signal subspace.

3. Perform SVD on the $M \times 2P$ matrix $\mathbf{E}_{12} \triangleq [\mathbf{E}_1 \ \mathbf{E}_2]$; place the last $P$ right singular vectors of $\mathbf{E}_{12}$ in $\mathbf{V}_2$; partition $\mathbf{V}_2$ into two $P \times P$ matrices $\mathbf{V}_{12}$ and $\mathbf{V}_{22}$, i.e., $\mathbf{V}_2 \triangleq [\mathbf{V}_{12}^T \ \mathbf{V}_{22}^T]^T$.

4. Perform eigenvalue decomposition on the $P \times P$ matrix $\Psi \triangleq -\mathbf{V}_{12}\mathbf{V}_{22}^{-1}$; let $\Phi_{pp}$, $p = 1, 2, \cdots, P$ be the eigenvalues. Note, from (3.7), that in the ideal case, where $r_{2y} = 0$,

$$\angle\Phi_{pp} = 2\pi r_{2x} \sin\phi_p - \angle g_2, \qquad p = 1, 2, \cdots, P. \tag{3.9}$$

5. Compute $\hat{\phi}_p$ using

$$\hat{\phi}_p = \arcsin\left(\frac{\angle\Phi_{pp} + \angle g_2}{2\pi r_{2x}}\right) \tag{3.10}$$

In the rest of this chapter, we assume that VESPA computes (see steps 1 to 4) the diagonal matrix $\Phi$ from $\mathbf{C}_1$ and $\mathbf{C}_2$ as long as $\mathbf{C}_1$ and $\mathbf{C}_2$ are described by (3.6)-(3.8), and the rank conditions are satisfied. The DOA information is then extracted by applying simple algebra [see (3.10)] to the diagonal entries of $\Phi$ based on the prior knowledge of the array information. To implement VESPA only requires prior knowledge of $\angle g_2$ and $r_{2x}$.

### 3.1.4 Perturbation Model

We have defined the model parameters for an arbitrary array to be sensor gains $\{g_i\}_{i=2}^M$, and, sensor locations $\{r_{ix}\}_{i=2}^P$ and $\{r_{iy}\}_{i=2}^P$. Let the perturbed value of $g_i$ be denoted by $\tilde{g}_i$, where

$$\tilde{g}_i \triangleq |\tilde{g}_i| e^{j\angle\tilde{g}_i} = (|g_i| + \varepsilon_{i1}) e^{j(\angle g_i + \varepsilon_{i2})}, \qquad i = 1, \cdots, M \tag{3.11}$$

in which $\varepsilon_{i1}$ and $\varepsilon_{i2}$ represent amplitude and phase perturbations on $g_i$, respectively.

One advantage of VESPA is that it does not have a geometrical constraint, i.e., the array configuration can be linear, planar, or even volumetric. In [13], the array is planar; our results are for such planar arrays. The perturbed sensor position $\tilde{r}_i$ is denoted by

$$\tilde{r}_{ix} = r_{ix} + \varepsilon_{ix} \qquad \text{and} \qquad \tilde{r}_{iy} = r_{iy} + \varepsilon_{iy} \qquad i = 1, \cdots, M \quad (3.12)$$

where $r_{ix}$ and $r_{iy}$ are the nominal positions of $r_i$ along the $x$- and $y$-axes, respectively, and, $\varepsilon_{ix}$ and $\varepsilon_{iy}$ are respective position perturbations along the $x$- and $y$-axes.

The characteristics of the real-valued perturbations, $\varepsilon$'s, can be either deterministic or random [92]. If $\varepsilon$'s are deterministic, they will introduce a *bias*, $\tilde{\phi}_p - \phi_p$, in the estimate of $\phi_p$. If $\varepsilon$'s are random, we can obtain a measure of the average effect of the model errors, measured in terms of *bias* and *standard deviation* (STD).

In practice, model perturbations may be caused by factors including gain errors, sensor position errors, mutual coupling between sensors, or receiver fluctuations due to temperature and humidity. Although the adopted model is simple, it is widely used throughout the performance analysis literature.

## 3.2 Sensitivity Analysis

In this section, we will continue the classical sensitivity analysis procedure to derive closed-form formulas of various sensitivities, with respect to individual or combined and deterministic or random perturbations. Because of the important roles of guiding sensors in VESPA, the derivations of sensitivities with respect to parameters of guiding sensors is separated from those of non-guiding sensors.

### 3.2.1 Perturbations on Sensor Gains $\{g_i\}_{i=2}^{M}$

Steps **[CS2]** through **[CS5]** of the classical sensitivity analysis procedure are now applied for determining individual deterministic sensitivities. In **[CS2]**, the (unperturbed) inputs to VESPA are $\mathbf{C}_1$ and $\mathbf{C}_2$, defined in (3.6). Since no perturbations occur, VESPA, naturally, arrives at the correct parameters $\{\phi_p\}_{p=1}^{P}$.

To derive the sensitivity with respect to *non-guiding sensor gains* $\{g_i\}_{i=3}^M$, we assume that only one sensor gain $\tilde{g}_i$, $i \in \{3,4,\cdots,M\}$, is perturbed, and the remaining $\{\tilde{g}_j\}_{j=3,j\neq i}^M$ are not perturbed. Replacing $g_i$ with $\tilde{g}_i$, $i \in \{3,\cdots,M\}$, in (3.6)-(3.8), we have

$$\tilde{\mathbf{C}}_1 = \tilde{\mathbf{A}}\Lambda\tilde{\mathbf{A}}^H \qquad \text{and} \qquad \tilde{\mathbf{C}}_2 = \tilde{\mathbf{A}}\Phi\Lambda\tilde{\mathbf{A}}^H. \tag{3.13}$$

The symbol $\tilde{\mathbf{A}}$ is used instead of $\mathbf{A}$, because the $i$th row of $\mathbf{A}$ has been changed due to the change of $g_i$ to $\tilde{g}_i$; however, the diagonal matrix $\Phi$, containing the DOA information, is not changed. Once we have perfect knowledge of $\tilde{\mathbf{C}}_1$ and $\tilde{\mathbf{C}}_2$ in (3.13), we can, through VESPA, obtain $\Phi$, and then $\{\phi_p\}_{p=1}^P$, i.e., $\tilde{\phi}_p = \phi_p$, $p = 1,\cdots,P$. This is because $\tilde{\phi}_p$ depends on $\angle g_2$, but it does not depend on $\tilde{g}_i$, $i \in \{3,4,\cdots,M\}$ [see (3.7)]. According to (3.1), we have the individual deterministic sensitivities,

$$S_{g_i}^{\phi_p} \triangleq \frac{\tilde{\phi}_p - \phi_p}{\phi_p} = 0, \qquad p = 1,2,\cdots,P \qquad i = 3,4,\cdots,M. \tag{3.14}$$

In practice, when finite-sample effect is considered, the change of $g_i$ does influence algorithm performance. In implementation, we can only compute estimates of $\tilde{\mathbf{C}}_1$ and $\tilde{\mathbf{C}}_2$ from finite samples of data. The estimation quality of $\tilde{\mathbf{C}}_1$ and $\tilde{\mathbf{C}}_2$ depends on the steering matrix $\tilde{\mathbf{A}}$ [101]. Although $\Phi$ does not change with respect to the change of $\{g_i\}_{i=3}^M$, the steering matrix now is $\tilde{\mathbf{A}}$; hence, the influence of perturbations must be considered if finite-sample effect is considered; or, the finite-sample effect can be ignored for high SNR cases. If both the finite-sample effect and model error are considered, we need to combine the work by Yuen and Friedlander [101] with our present work.

To derive the sensitivity with respect to *guiding sensor complex gain*, we assume only $g_2$ is perturbed, and $\{g_i\}_{i=3}^M$ are not perturbed. Replacing $g_2$ with $\tilde{g}_2$ in (3.6)-(3.8) and letting $r_{2y} = 0$, we have

$$\tilde{\mathbf{C}}_1 = \tilde{\mathbf{A}}\Lambda\tilde{\mathbf{A}}^H \qquad \text{and} \qquad \tilde{\mathbf{C}}_2 = \tilde{\mathbf{A}}\tilde{\Phi}\Lambda\tilde{\mathbf{A}}^H \tag{3.15}$$

where

$$\tilde{\Phi} \triangleq |\tilde{g}_2| \ \mathrm{diag}\{e^{j(2\pi r_{2x}\sin\phi_1 - \angle\tilde{g}_2)}, e^{j(2\pi r_{2x}\sin\phi 2 - \angle\tilde{g}_2)}, \cdots, e^{j(2\pi r_{2x}\sin\phi_P - \angle\tilde{g}_2)}\} \qquad (3.16)$$

The perturbation $\tilde{g}_2$ causes changes of $\mathbf{A}$ and $\Phi$ to $\tilde{\mathbf{A}}$ and $\tilde{\Phi}$, respectively. Note that the $\tilde{\mathbf{A}}$ in (3.15) is different from the $\tilde{\mathbf{A}}$ in (3.13). We use the same notation only to indicate that $\tilde{\mathbf{A}}$ is perturbed from $\mathbf{A}$. Regardless of the change in $\mathbf{A}$, VESPA is able to compute $\tilde{\Phi}$ shown in (3.16).

VESPA (or ESPRIT) does not know that $\angle g_2$ has changed to $\angle\tilde{g}_2$; so it still computes $\phi_p$ as in (3.10). Since this computation uses $\angle\tilde{\Phi}_{pp}$, we denote the computed value of $\phi_p$ as $\tilde{\phi}_p$, i.e., $\tilde{\phi}_p = \arcsin[(\angle\tilde{\Phi}_{pp} + \angle g_2)/2\pi r_{2x}]$. From (3.16), we know that $\angle\tilde{\Phi}_{pp} = 2\pi r_{2x}\sin\phi_p - \angle\tilde{g}_2$, which can be substituted into $\tilde{\phi}_p$, so that we obtain the final result,

$$\tilde{\phi}_p = \arcsin(\sin\phi_p + \frac{\angle g_2 - \angle\tilde{g}_2}{2\pi r_{2x}}), \qquad p = 1, \cdots, P. \qquad (3.17)$$

Substituting (3.17) into (3.1), and combining the result with (3.14), we have the individual deterministic sensitivity

$$S_{g_i}^{\phi_p} = \begin{cases} \frac{1}{\phi_p}[\arcsin(\sin\phi_p + \frac{\angle g_2 - \angle\tilde{g}_2}{2\pi r_{2x}}) - \phi_p], & i = 2 \quad \text{and} \quad p = 1, 2, \cdots, P \\ 0, & i = 3, \cdots, M \quad \text{and} \quad p = 1, \cdots, P \end{cases} \qquad (3.18)$$

The situation of combined (multiple) deterministic perturbations is considered next. From (3.6)-(3.8), each perturbed $\tilde{g}_i, i = 2, \cdots, M$, changes the $i$th row of $\mathbf{A}$, and, moreover, only $\tilde{g}_2$ changes $\Phi$. If multiple perturbations occur, the two cumulant matrices are still related by the formulas in (3.15), with a different $\tilde{\mathbf{A}}$; but, $\tilde{\Phi}$ is still described by (3.16). If these perturbations do not cause a rank reduction of $\tilde{\mathbf{A}}$, then we can still use VESPA to solve for the perturbed $\tilde{\Phi}$, which in turn gives $\{\tilde{\phi}_p\}_{p=1}^{P}$. Since non-guiding sensors do not contribute to $\tilde{\Phi}$, the combined effect is equal to the individual effects in (3.18), i.e., $(\tilde{\phi}_p - \phi_p)/\phi_p$ still has the same expression as in the first line of (3.18).

We conclude, from (3.18), that only the phase of the guiding sensor $\tilde{g}_2$ causes $\tilde{\phi}_p$ to deviate from $\phi_p$; the amplitude of $\tilde{g}_2$ has no effect. Observe that if $\angle\tilde{g}_2 = \angle g_2$, then $S_{g_i}^{\phi_p} = 0$ for $\forall i$ and $\forall p$.

### 3.2.2 Perturbations on Sensor Positions $\{r_{2x}\}_{i=2}^{M}$ and $\{r_{2y}\}_{i=2}^{M}$

The sensitivity derivations for sensor position perturbations are analogous to those given for sensor gain perturbations. As in the previous subsection, sensitivities with respect to parameters of non-guiding sensors are derived separately from those for guiding sensors. For deterministic perturbations on *non-guiding sensor positions*, replacing $r_{ix}$ with $\tilde{r}_{ix}$, and $r_{iy}$ with $\tilde{r}_{iy}$, $i \in \{3, \cdots, M\}$ in (3.6)-(3.8), we have the same perturbed equations as (3.13) but with a different $\tilde{\mathbf{A}}$. The sensitivities with respect to positions of non-guiding sensors equals zero.

For *guiding sensor positions*, replacing $r_{2x}$ with $\tilde{r}_{2x}$, and $r_{2y}$ with $\tilde{r}_{2y}$ in (3.6)-(3.8), we have the same perturbed equations, (3.15) and (3.16), but, again, with a different $\tilde{\mathbf{A}}$, and a different

$$\tilde{\Phi} \triangleq |g_2| \operatorname{diag}\{e^{j[2\pi(\tilde{r}_{2x}\sin\phi_1 + \tilde{r}_{2y}\cos\phi_1) - \angle g_2]}, \cdots, e^{j[2\pi(\tilde{r}_{2x}\sin\phi_P + \tilde{r}_{2y}\cos\phi_P) - \angle g_2]}\}. \quad (3.19)$$

VESPA (or ESPRIT) does not know that $r_{2x}$ and $r_{2y}$ have changed to $\tilde{r}_{2x}$ and $\tilde{r}_{2y}$; so, it still computes $\phi_p$ as in (3.10), and, we denote this value as $\tilde{\phi}_p$, i.e., $\tilde{\phi}_p = \arcsin[(\angle\tilde{\Phi}_{pp} + \angle g_2)/(2\pi r_{2x})]$. From (3.19), we know that $\angle\tilde{\Phi}_{pp} = 2\pi(\tilde{r}_{2x}\sin\phi_p + \tilde{r}_{2y}\cos\phi_p) - \angle g_2$, which when substituted into $\tilde{\phi}_p$, gives us

$$\tilde{\phi}_p = \arcsin(\frac{\tilde{r}_{2x}}{r_{2x}}\sin\phi_p + \frac{\tilde{r}_{2y}}{r_{2x}}\cos\phi_p), \qquad p = 1, 2, \cdots, P. \quad (3.20)$$

Substituting (3.20) into (3.1), and combining it with the results for positions of non-guiding sensors, we have the individual deterministic sensitivity

$$S_{r_i}^{\phi_p} = \begin{cases} \frac{1}{\phi_p}[\arcsin(\frac{\tilde{r}_{2x}}{r_{2x}}\sin\phi_p + \frac{\tilde{r}_{2y}}{r_{2x}}\cos\phi_p) - \phi_p], & i = 2 \quad \text{and} \quad p = 1, 2, \cdots, P \\ 0, & i = 3, \cdots, M \quad \text{and} \quad p = 1, \cdots, P \end{cases} \quad (3.21)$$

If $\tilde{r}_{2x} = r_{2x}$, and because $\tilde{r}_{2y} = r_{2y} = 0$ (see assumption **[AS4]**), then $\tilde{\phi}_p = \phi_p$, in which case $S_{r_i}^{\phi_p} = 0, \forall i$.

For the same reason given in the preceding subsection, the combined sensitivity, when multiple deterministic perturbations occur to $\{r_{ix}\}_{i=2}^{M}$ and to $\{r_{iy}\}_{i=2}^{M}$, is also given by (3.21). We conclude, from (3.21), that only sensor positions of guiding sensors $r_{2x}$ and $r_{2y}$ affect the sensitivity of VESPA.

### 3.2.3 Multiple Perturbations on Both Sensor Gains and Sensor Positions

If all parameters are perturbed, then, by combining (3.17) and (3.20), we obtain

$$\tilde{\phi}_p = \arcsin\left(\frac{\tilde{r}_{2x}}{r_{2x}} \sin\phi_p + \frac{\tilde{r}_{2y}}{r_{2x}} \cos\phi_p + \frac{\angle g_2 - \angle \tilde{g}_2}{2\pi r_{2x}}\right), \qquad p = 1, 2, \cdots, P. \tag{3.22}$$

It can be concluded from (3.22) that only sensor gain and sensor positions of the guiding sensor influence the sensitivity of estimated DOAs. Observe that, if $\tilde{r}_{2x} = r_{2x}$, $\tilde{r}_{2y} = r_{2y} = 0$, and $\angle \tilde{g}_2 = \angle g_2$, then (3.22) reduces to the identity that $\tilde{\phi}_p = \phi_p$, as it should.

When the sensor perturbations are random, means and standard deviations of the random sensitivities are computed by taking ensemble means and standard deviations of (3.18) and (3.21). The sensitivities for the situation when all sensor gains and sensor positions are perturbed randomly can be computed similarly.

Our closed-form sensitivity formulas for VESPA are surprisingly simple. It is very difficult to obtain comparable formulas for ESPRIT, because ESPRIT requires two completely identical arrays, whereas VESPA only requires one identical pair of sensors.

## 3.3 Small Signal Analysis

The complicated sensitivity formulas of (3.18) and (3.21) are difficult to use to draw conclusions about the influence of perturbations. When calibration is applied for the measurement data snapshots, only small model mismatches exist or small

perturbations occur to model parameters. For these reasons, we apply first-order approximation to (3.18) and (3.21), and obtain small-signal formulas for the sensitivity of interest.

Using the first-order Taylor series approximation of $\arcsin(x)$ around $x_0 \in [0,1]$,

$$\arcsin(x) \approx \arcsin(x_0) + \frac{x - x_0}{\sqrt{1 - x_0^2}}, \tag{3.23}$$

with $x = \sin\phi_p + (\angle g_2 - \angle \tilde{g}_2)/(2\pi r_{2x})$ and $x_0 = \sin\phi_p$ in (3.18), we have

$$\begin{aligned} S_{g_i}^{\phi_p} &\approx \begin{cases} \frac{1}{\phi_p}[\arcsin(\sin\phi_p) + \frac{\angle g_2 - \angle \tilde{g}_2}{\sqrt{1-\sin^2\phi_p}} \frac{1}{2\pi r_{2x}} - \phi_p] \\ 0, \end{cases} \\ &= \begin{cases} \frac{1}{2\pi r_{2x}\phi_p \cos\phi_p}(\angle g_2 - \angle \tilde{g}_2) & i = 2 \quad \text{and} \quad p = 1,2,\cdots,P \\ 0, & i = 3,4,\cdots,M \quad \text{and} \quad p = 1,2,\cdots,P \end{cases} \end{aligned} \tag{3.24}$$

assuming $\phi_p \in [0°, 90°]$. With the assumption that $\phi_p \in [0°, 90°]$, we do not have to consider the sign of $\sqrt{1 - x_0^2}$ in (3.23) and the sign of $\cos\phi_p$ in (3.24); the derivation is, therefore, greatly simplified. The generalization to non-positive $x_0$ is straightforward; but, in the following, $x_0$ is assumed to be positive.

Applying the approximation formula (3.23) with $x = \frac{\tilde{r}_{2x}}{r_{2x}}\sin\phi_p + \frac{\tilde{r}_{2y}}{r_{2x}}\cos\phi_p$ and $x_0 = \sin\phi_p$ to (3.21), we have

$$\begin{aligned} S_{r_i}^{\phi_p} &\approx \begin{cases} \frac{1}{\phi_p}[\arcsin(\sin\phi_p) + \frac{\frac{\tilde{r}_{2x}}{r_{2x}}\sin\phi_p + \frac{\tilde{r}_{2y}}{r_{2x}}\cos\phi_p - \sin\phi_p}{\sqrt{1-\sin^2\phi_p}} - \phi_p] \\ 0, \end{cases} \\ &= \begin{cases} \frac{1}{\phi_p}[\frac{\tilde{r}_{2y}}{r_{2x}} + (\frac{\tilde{r}_{2x}}{r_{2x}} - 1)\tan\phi_p], & i = 2, \quad \text{and}, \quad p = 1,2,\cdots,P \\ 0, & i = 3,4,\cdots,M, \quad \text{and}, \quad p = 1,2,\cdots,P \end{cases} \end{aligned} \tag{3.25}$$

where $\sin\phi_p$ and $\cos\phi_p$ are assumed to be positive. As in (3.18) and (3.21), (3.24) and (3.25) are for individual and combined deterministic sensitivities.

## 3.4 Experiments

### 3.4.1 Simulation Setup

The sensitivity formulas were derived by assuming that the cumulant matrices are only contaminated by model perturbations. In practice, cumulant matrices are unknown and need to be estimated from data snapshots. To run computer simulations, instead of considering $S_{g_i}^{\phi_p}$, we consider

$$S_{g_i}^{\hat{\phi}_p} \triangleq \frac{\hat{\tilde{\phi}}_p - \hat{\phi}_p}{\hat{\phi}_p}, \qquad p = 1, 2, \cdots, P \quad \text{and} \quad i = 2, \cdots, M. \tag{3.26}$$

where $\hat{\phi}_p$ is the estimate of $\phi_p$ and $\hat{\tilde{\phi}}_p$ is the estimate of $\tilde{\phi}_p$, both of which are obtained from VESPA.

The relationship between $S_{g_i}^{\phi_p}$ and $S_{g_i}^{\hat{\phi}_p}$ is derived below. Let the estimate $\hat{\phi}_p$ of $\phi_p$ be defined as

$$\hat{\phi}_p \triangleq \phi_p + e_{\phi_p} \tag{3.27}$$

where $e_{\phi_p}$ is the estimation error. Substituting (3.27) into (3.26), we have

$$S_{g_i}^{\hat{\phi}_p} = \frac{\hat{\tilde{\phi}}_p - \phi_p - e_{\phi_p}}{\phi_p + e_{\phi_p}} \tag{3.28}$$

Adding and subtracting $\tilde{\phi}_p$ from the numerator, and rearranging the terms, we have

$$S_{g_i}^{\hat{\phi}_p} = \frac{\tilde{\phi}_p - \phi_p}{\phi_p} \quad \frac{1 + (\hat{\tilde{\phi}}_p - \tilde{\phi}_p - e_{\phi_p})/(\tilde{\phi}_p - \phi_p)}{1 + e_{\phi_p}/\phi_p} \tag{3.29}$$

The first term on the RHS of (3.29) is $S_{g_2}^{\phi_p}$; so, we finally have

$$S_{g_i}^{\hat{\phi}_p} = S_{g_i}^{\phi_p} \quad \frac{1 + (\hat{\tilde{\phi}}_p - \tilde{\phi}_p - e_{\phi_p})/(\tilde{\phi}_p - \phi_p)}{1 + e_{\phi_p}/\phi_p} \tag{3.30}$$

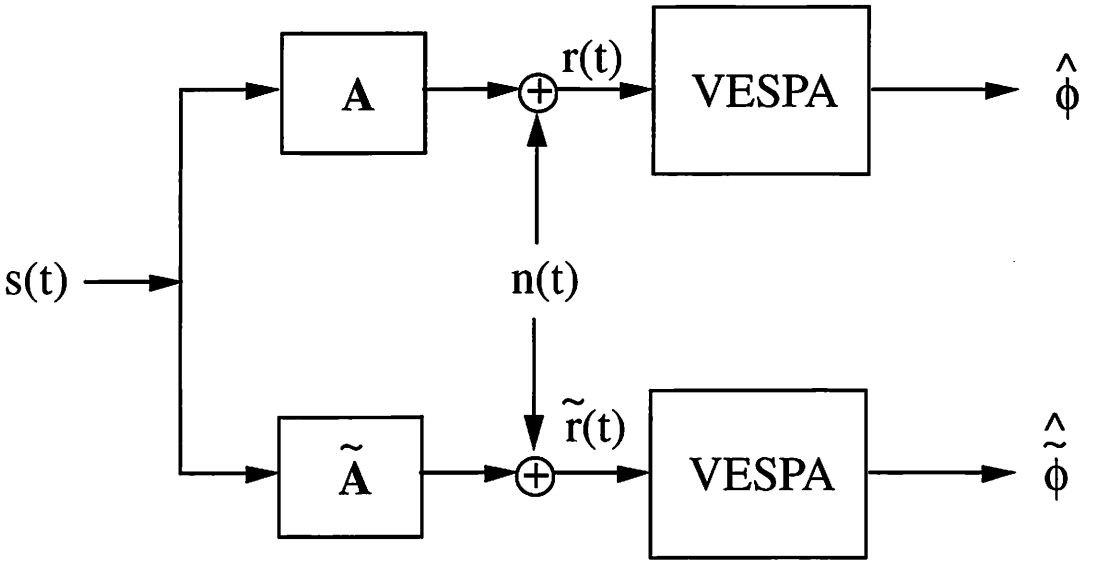



Figure 3.1: Simulation model: $\mathbf{s}(t)$ is a zero-mean random vector denoting the source vector; $\mathbf{A}$ and $\tilde{\mathbf{A}}$ are the nominal steering matrix and perturbed steering matrix, respectively, that depend on the array configurations; and, $\mathbf{n}(t)$ is the additive white Gaussian noise.

By setting $e_{\phi_p} = 0$ in (3.30), the situation when perfect estimation is available; then, $\hat{\tilde{\phi}}_p = \tilde{\phi}_p$ and (3.30) reduces to $S_{g_i}^{\phi_p}$. For high SNR and long data snapshots, $e_{\phi_p} \approx 0$, and $S_{g_i}^{\hat{\phi}_p} \approx S_{g_i}^{\phi_p}$.

The experimental setup is depicted in Fig. 3.1. In the following experiments, three independent sources with DOAs $\phi_1 = 14°$, $\phi_2 = 23°$, and $\phi_3 = 40°$ illuminate the array. All sources have the same power, and, therefore, all three sources have the same SNR. Each signal is sampled from a phase-modulated (PM) signal. The arrays used are depicted in Fig. 3.2. Guiding sensors for VESPA(5) and VESPA(8) are chosen to be $r_1$ and $r_2$; and the *virtual* copy of the main array is shown. The nominal sensor gain for each sensor is assumed to be unity, i.e., $g_i = 1$, $i \in \{1, 2, \cdots, M\}$. Nominal positions of the sensors satisfy the constraints in Section 3.1.2.

All experiments assume that random perturbations occur to either all sensor gains, $\{g_2\}_{i=2}^{M}$, or all sensor positions, $\{r_{ix}, r_{iy}\}_{i=2}^{M}$. All perturbations are *i.i.d.* and are uniformly distributed over $[-\varepsilon/2, \varepsilon/2]$, denoted by UNIF$(-\varepsilon/2, \varepsilon/2)$. Sample means and sample STDs of $\{(\hat{\tilde{\phi}}_p - \hat{\phi}_p)/\hat{\phi}_p\}_{p=1}^{P}$ are computed for verification of the sensitivity formulas.

Section 3.4.2 verifies the sensitivity formulas derived in this chapter. Sections 3.4.3-3.4.5 are for the more general robustness questions where moderate or severe noise, finite samples, and model perturbations are considered.

### 3.4.2 Sensitivities of VESPA(5)

A double-nested Monte Carlo (MC) simulation was constructed for VESPA(5). First, at fixed $\varepsilon = 0.01$, for which $\{\varepsilon_{i1}, \varepsilon_{i2}\}_{i=2}^{M}$ are *i.i.d.* UNIF$[-0.005, 0.005]$, we ran 50 MC iterations over $\{\tilde{g}_i\}_{i=2}^{M}$; and, within each MC iteration, we performed 50 MC sub-iterations over 2000 snapshots of signals and measurement noise for fixed SNR = 20 dB. For each $\varepsilon$ and the fixed SNR = 20 dB with 2000 snapshots for each MC iteration, 2500 MC iterations were conducted. The results, plotted in Fig. 3.3, are for $\varepsilon = 0.01, 0.02, \cdots, 0.2$ and required a total of 50,000 simulations.

Experimental results and analytical results, the latter obtained by computing the ensemble mean and standard deviation of (3.18), are provided in Fig. 3.3 for comparison. Note that the ensemble mean and standard deviation of (3.18) are computed by treating the $\angle g_2 - \angle \tilde{g}_2$ on the RHS as the only random variable, which is uniform over $[-\varepsilon/2, \varepsilon/2]$. The analytical means of normalized sensitivities for $\phi_1$, $\phi_2$, and $\phi_3$ for $\varepsilon = 0.2$ are between $1.7 \times 10^{-4}$ and $3.5 \times 10^{-4}$, which are difficult to observe from Fig. 3.3(a); however, we can see the experimental sample means distributed randomly around 0. The sample STDs [see Fig. 3.3(b)] are very close to their analytical values. Similar experiments were conducted for perturbations on sensor positions $\{r_{ix}, r_{iy}\}_{i=2}^{M}$. The experimental results, illustrated in Fig. 3.4, agree with their analytical values.

The sensitivity formulas (3.18) and (3.21) are functions of only guiding sensor parameters, i.e., $g_2$, $r_{2x}$, and $r_{2y}$; hence, VESPA(5) and VESPA(8) have the same sensitivities since they use the same guiding sensors.

The analytical curves in Fig. 3.3(b) resemble their counterparts in 3.4(b), both of which are "roughly" straight lines with different slopes. This can be explained in terms of the so-called spatial frequencies $\{\angle\Phi_{pp}\}_{p=1}^{P}$ [see (3.7)]. If only perturbation $\varepsilon_{22}$ (radians) occurs to $\angle g_2$, then each $\angle\tilde{\Phi}_{pp}$ is offset from $\angle\Phi_{pp}$ by $\varepsilon_{22}$ (radians). If respective perturbations, $\varepsilon_{2x}$ and $\varepsilon_{2y}$ (wavelengths), occur to $r_{2x}$ and $r_{2y}$, then $\angle\tilde{\Phi}_{pp}$ deviates from $\angle\Phi_{pp}$ by $2\pi\sqrt{\varepsilon_{2x}^2 \sin^2\phi_p + \varepsilon_{2y}^2 \cos^2\phi_p}$ (radians). Note that $\varepsilon_{2x}$ is

dimensionless. We selected $\varepsilon_{2x} = \varepsilon_{2y}$ in this experiment, which caused a $2\pi\varepsilon_{2x}$ (radians) deviation for $\{\angle\tilde{\Phi}_{pp}\}_{p=1}^{P}$, that, in turn, caused perturbations to DOA estimates. Consequently, the perturbation of $\varepsilon_{22}$ to $\angle g_2$ is equivalent to the perturbation of $2\pi\varepsilon_{2x}$ in both $r_{2x}$ and $r_{2y}$. This explains why the analytical curves in Fig. 3.4(b) are $2\pi$ scaled versions of those in Fig. 3.3(b).

### 3.4.3 Comparison of VESPA(8) and ESPRIT(8) in the Presence of Model Errors

In this experiment, we compare the normalized sensitivities of VESPA(8) and ESPRIT(8). We repeated the experiment in Section 3.4.2 for ESPRIT(8), and plot the results for $\phi_1$, along with those for VESPA(8), in Figs. 3.5 and 3.6.

Figure 3.5 compares the normalized sensitivities of VESPA(8) and ESPRIT(8) in the presence of multiple random sensor gains; and, Fig. 3.6 depicts the case of multiple random perturbations on sensor positions. Obviously, in both figures, VESPA(8) has smaller sample means and smaller sample STDs of $(\hat{\tilde{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$ than those of ESPRIT(8). Although not provided here, the results for $\phi_2$ and $\phi_3$ yielded the same observations.

Combining this conclusion with the one in Section 3.4.2, that VESPA(5) and VESPA(8) have the same sensitivities, we conclude that VESPA(5) outperforms ESPRIT(8) in hardware savings (see [13]), and is more *robust to model perturbations.*

### 3.4.4 Comparison of VESPA(5) and ESPRIT(8) in the Presence of Finite Samples and Model Errors

We have shown that VESPA(5) and VESPA(8) are both more robust to model perturbations than ESPRIT(8). In Sections 3.4.2 and 3.4.3, finite-sample effect has been suppressed by setting SNR = 20 dB and the number of snapshots equal to 2000. In this experiment, we are interested in the situation when both finite samples and model perturbations exist.

A similar double MC simulation was conducted: fixed SNR = 20 dB, fixed $\varepsilon$ = 0.04, and the number of snapshots equal to $N$, ranging from 60 to 500 in steps of

4 snapshots, 50 MC iterations over $\{g_i\}_{i=2}^{8}$, and, within each MC iteration, 50 sub-iterations over signals and noise. A total of 277,500 simulations were conducted. Only the lower path of Fig. 3.1 was conducted, since finite samples and model errors simultaneously contribute to $\{\hat{\tilde{\phi}}_p\}_{p=1}^{3}$, whereas only finite samples contribute to $\{\hat{\phi}_p\}_{p=1}^{3}$.

Sample means and sample STDs of $\hat{\tilde{\phi}}_1 - \phi_1$ are plotted in Fig. 3.7 for different numbers of snapshots $N$. In Fig. 3.7(a), for $N > 100$, we observe that VESPA(5) has a smaller sample mean than ESPRIT(8), indicating that VESPA(5) produces less bias on the output estimates than ESPRIT(8). In Fig. 3.7(b), for $N < 120$, we observe that VESPA(5) suffers seriously from finite-sample effects, and has a much larger sample STD than ESPRIT(8); for $N > 250$, the model-error effects are worse for ESPRIT(8); for $120 < N < 250$, the combination of finite samples and model errors has about the same effects on VESPA(5) and ESPRIT(8).

We conclude from this experiment that in the presence of both finite samples and model errors, as long as "enough" snapshots are available, VESPA outperforms ESPRIT, in terms of smaller sample mean and sample STD of the estimation deviation from its true value.

### 3.4.5 Comparison of VESPA(5) and ESPRIT(8) in the Presence of Model Errors and Low SNRs

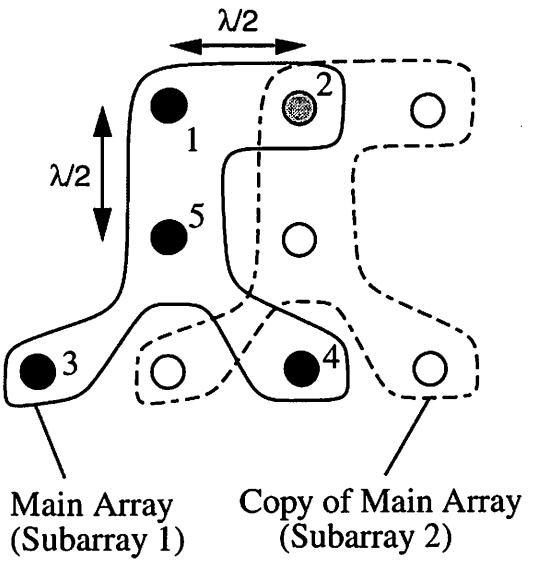
Another situation to evaluate the influence of finite samples is to consider the case of medium or low SNRs. To this end, we repeated the experiment in Section 3.4.3 for VESPA(5) and ESPRIT(8), except that SNR = 5 dB and 0 dB. Statistics of $\hat{\tilde{\phi}}_1 - \phi_1$ are plotted versus perturbations for SNR = 5 dB in Fig. 3.8 and SNR = 0 dB in Fig. 3.9.

It is clear from Figs. 3.8(a) and 3.9(a) that VESPA(5) has smaller bias than ESPRIT(8), and from Figs. 3.8(b) and 3.9(b) that VESPA(5) also has smaller STDs. This indicates that VESPA(5) is more robust to model errors than ESPRIT(8) even for medium or low SNRs.

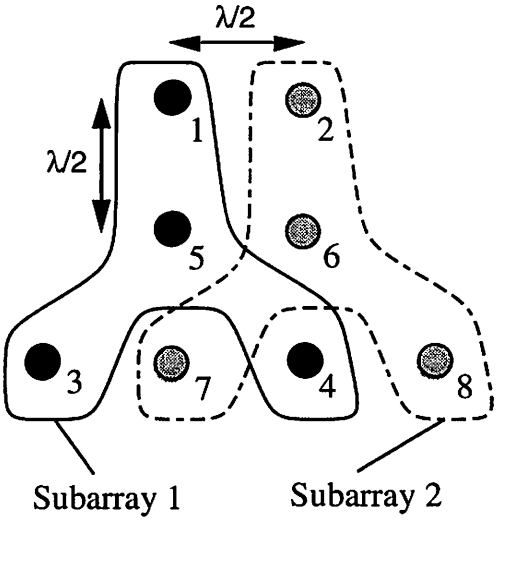
## 3.5 Conclusions

In this chapter, we followed the classical sensitivity procedure to analyze the sensitivity of VESPA in the presence of model perturbations. Closed forms of sensitivity formulas were derived. Computer simulations were conducted to verify the correctness of all derived formulas. Finally, we compared the sensitivities of VESPA(5) and ESPRIT(8), and concluded that VESPA(5) not only outperforms ESPRIT(8), in the already known sense of hardware savings, but is more robust to the general situations when model perturbations are present and only noisy measurements are available.

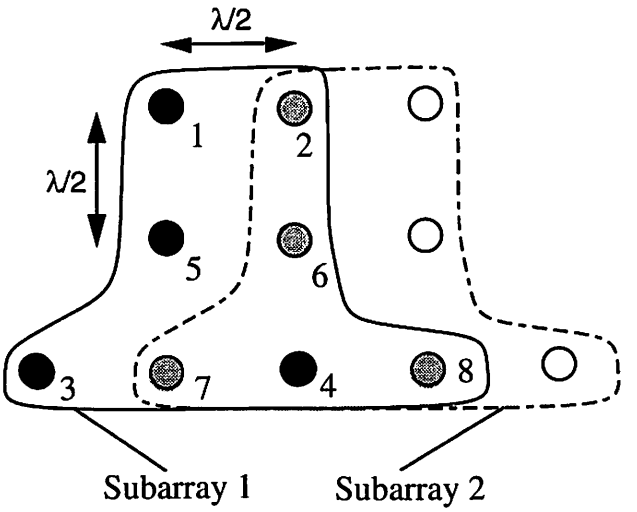
Although our results are for one array configuration and one DOA scenario, our methodology can be applied to arbitrary arrays and DOA scenarios, to study the general applicability of our conclusions.




(a) VESPA(5) Configuration

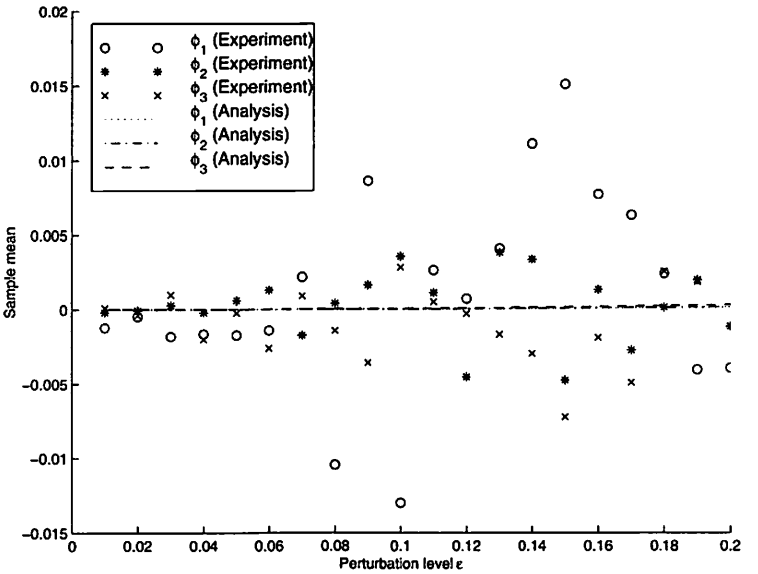



(b) ESPRIT(8) Configuration

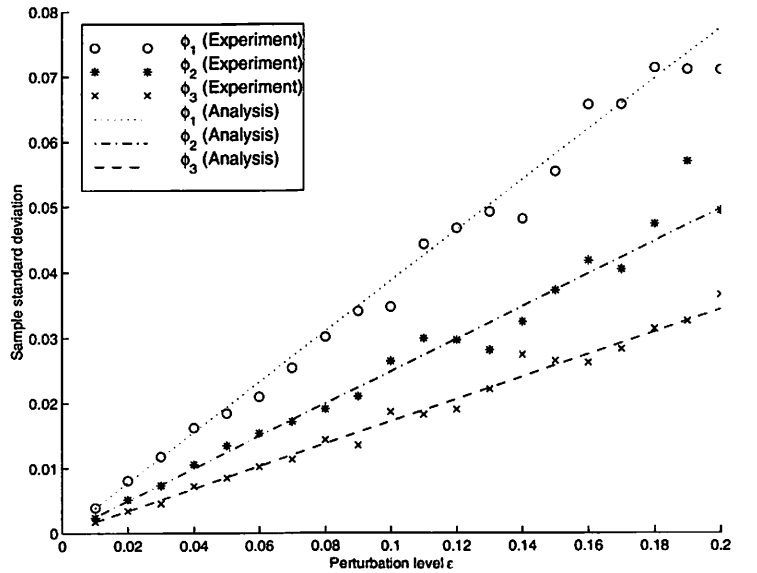



(c) VESPA(8) Configuration

Figure 3.2: Array configurations for (a) VESPA(5), (b) ESPRIT(8), and (c) VESPA(8). Circles with filled or shaded interiors denote actual sensors, and circles with empty interiors denote virtual sensors.

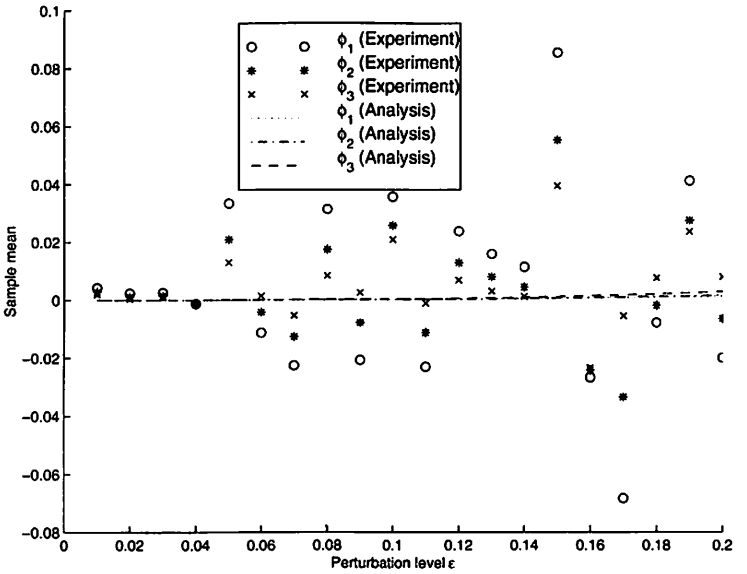



(a)

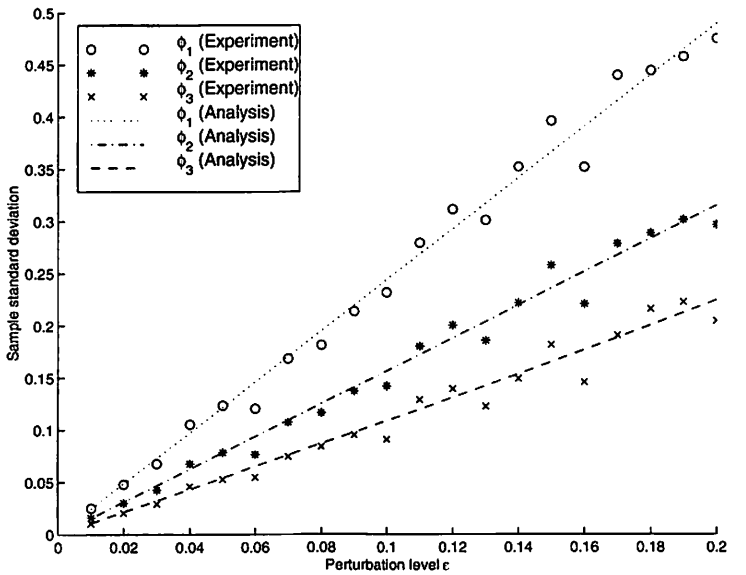



(b)

Figure 3.3: Experimental and analytical normalized sensitivities of VESPA(5) in the presence of multiple random perturbations on $\{g_i\}_{i=2}^{5}$, for SNR = 20 dB: (a) ensemble means of $\{(\tilde{\phi}_p - \phi_p)/\phi_p\}_{p=1}^{3}$ and sample means of $\{(\hat{\tilde{\phi}}_p - \hat{\phi}_p)/\hat{\phi}_p\}_{p=1}^{3}$; (b) ensemble STDs of $\{(\tilde{\phi}_p - \phi_p)/\phi_p\}_{p=1}^{3}$ and sample STDs of $\{(\hat{\tilde{\phi}}_p - \hat{\phi}_p)/\hat{\phi}_p\}_{p=1}^{3}$.

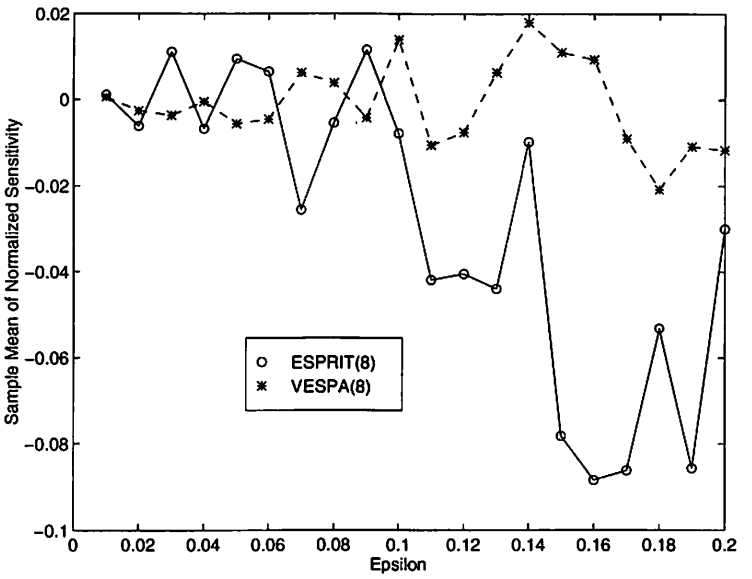



(a)

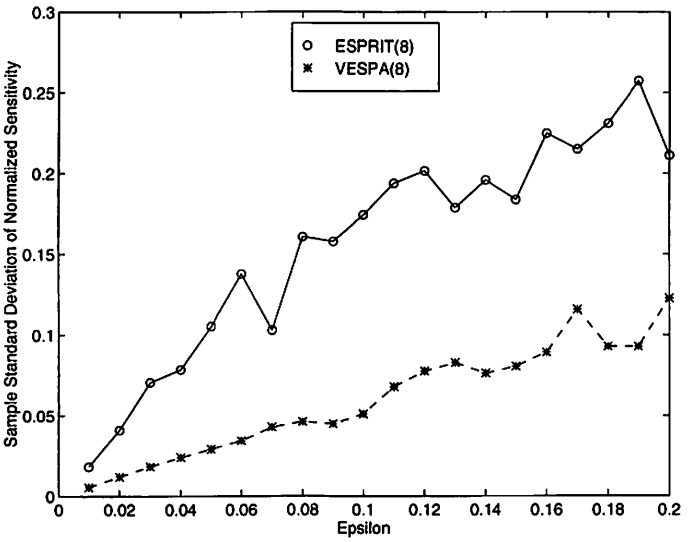



(b)

Figure 3.4: Experimental and analytical normalized sensitivities of VESPA(5) in the presence of multiple random perturbations on $\{r_{ix}, r_{iy}\}_{i=2}^{5}$ and SNR = 20 dB: (a) means of $\{(\tilde{\phi}_p - \phi_p)/\phi_p\}_{p=1}^{3}$ and sample means of $\{(\hat{\tilde{\phi}}_p - \hat{\phi}_p)/\hat{\phi}_p\}_{p=1}^{3}$; (b) ensemble STDs of $\{(\tilde{\phi}_p - \phi_p)/\phi_p\}_{p=1}^{3}$ and sample STDs of $\{(\hat{\tilde{\phi}}_p - \hat{\phi}_p)/\hat{\phi}_p\}_{p=1}^{3}$.

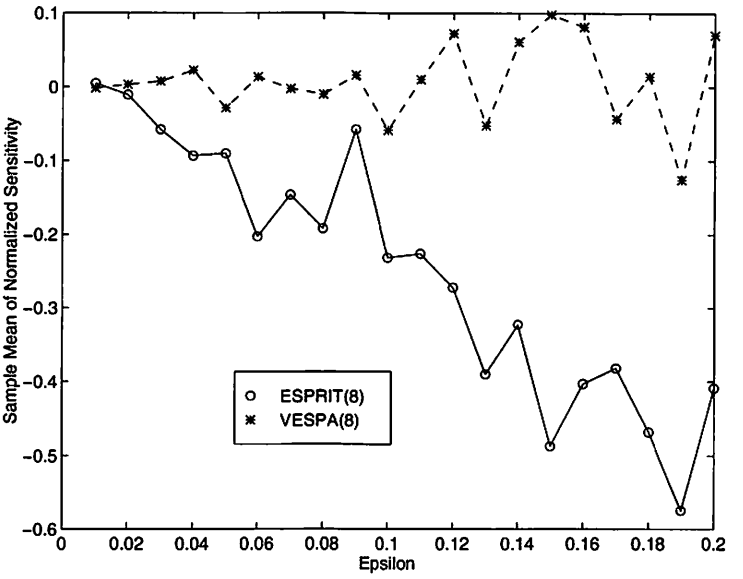



(a) Sample mean of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

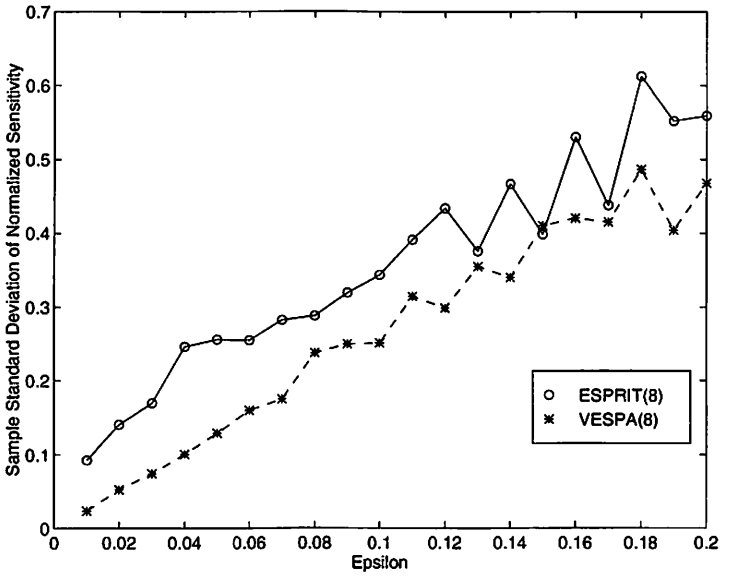



(b) Sample STD of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

Figure 3.5: Statistics of Normalized sensitivities of VESPA(8) and ESPRIT(8) in the presence of random perturbations on $\{g_i\}_{i=2}^{8}$ when SNR = 20 dB: (a) Sample mean of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$; (b) Sample STD of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$;

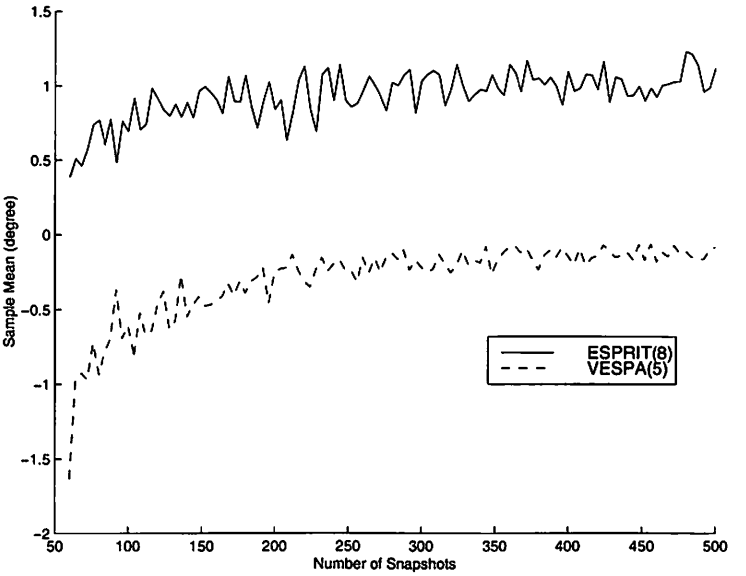



(a) Sample mean of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

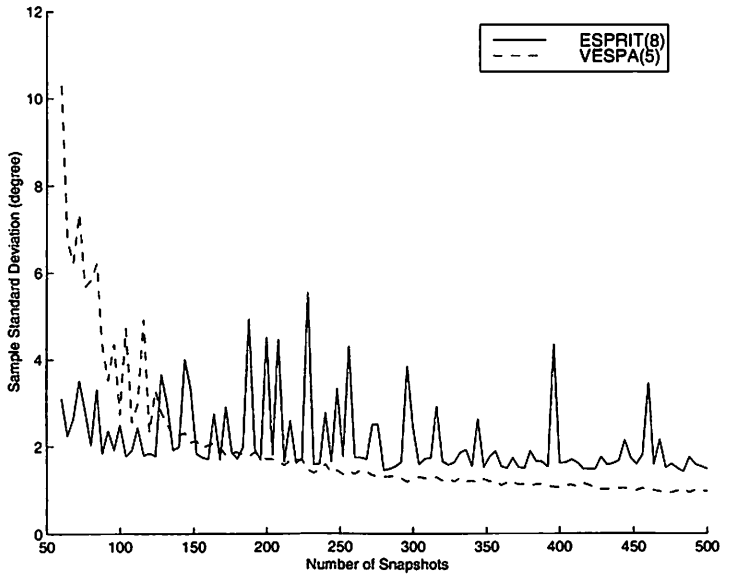



(b) Sample STD of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

Figure 3.6: Statistics of Normalized sensitivities of VESPA(8) and ESPRIT(8) in the presence of random position perturbations on $\{r_{ix}\}_{i=2}^{8}$ and $\{r_{iy}\}_{i=2}^{8}$ when SNR = 20 dB: (a) Sample mean of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$; (b) Sample STD of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$.

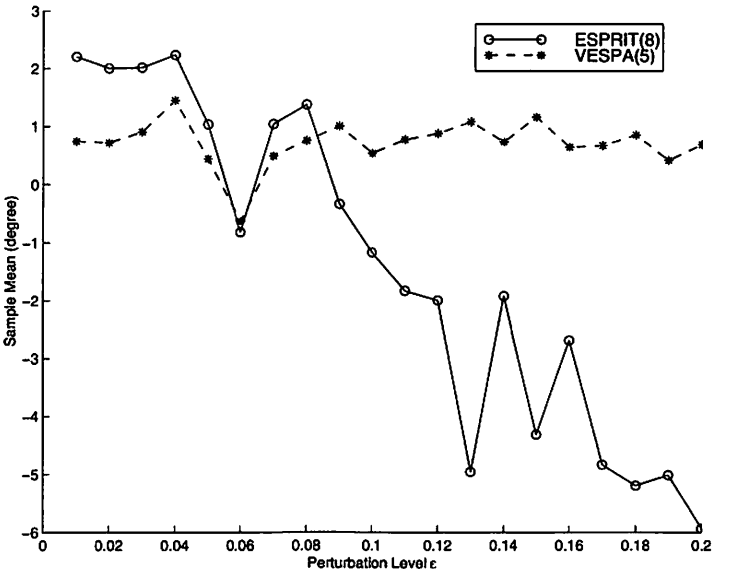



(a) Sample mean of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

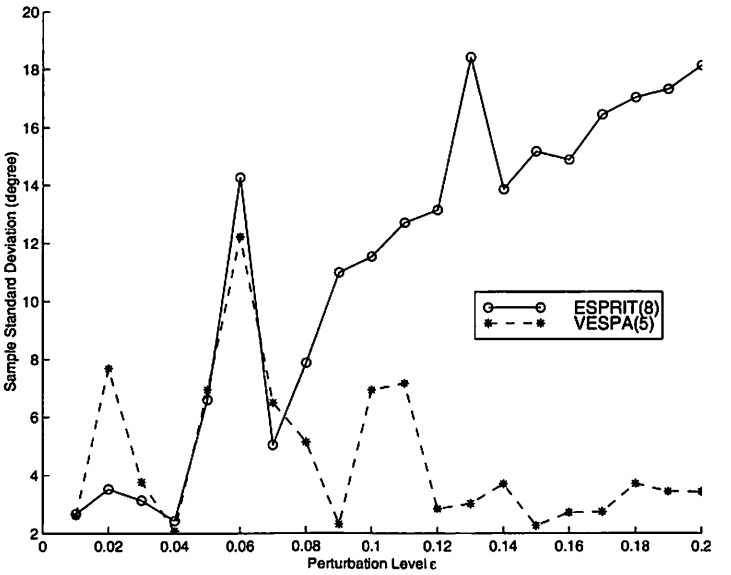



(b) Sample STD of $(\tilde{\hat{\phi}}_1 - \hat{\phi}_1)/\hat{\phi}_1$

Figure 3.7: Comparison of VESPA(5) and ESPRIT(8) in the presence of multiple perturbations on $\{g_i\}_{i=2}^{8}$ with $\varepsilon = 0.04$ and SNR = 20 dB, for different numbers of snapshots: (a) sample mean of $\tilde{\hat{\phi}}_1 - \phi_1$; (b) sample STD of $\tilde{\hat{\phi}}_1 - \phi_1$.

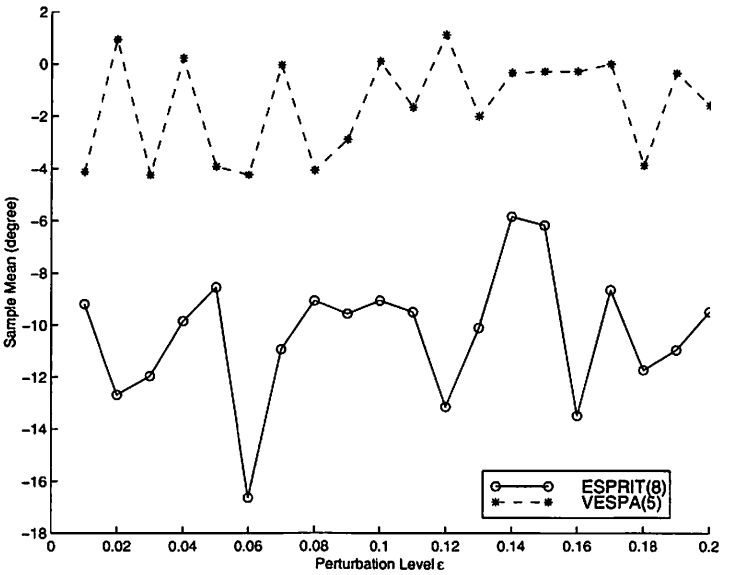



(a) Sample mean of $\hat{\tilde{\phi}}_1 - \hat{\phi}_1$

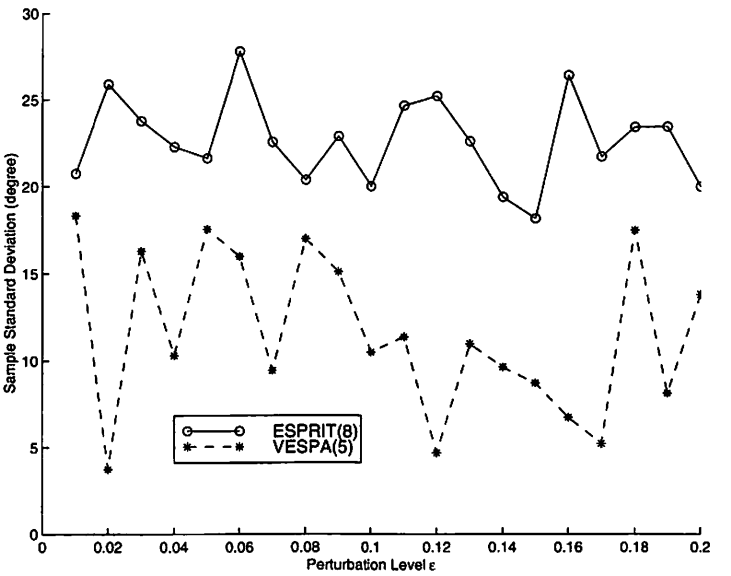



(b) Sample STD of $\hat{\tilde{\phi}}_1 - \hat{\phi}_1$

Figure 3.8: Comparison of VESPA(5) and ESPRIT(8) in the presence of random perturbations on $\{g_i\}_{i=2}^{8}$ and SNR = 5 dB: (a) sample mean of $\hat{\tilde{\phi}}_1 - \phi_1$; (b) sample STD of $\hat{\tilde{\phi}}_1 - \phi_1$.




(a) Sample mean of $\hat{\tilde{\phi}}_1 - \hat{\phi}_1$




(b) Sample STD of $\hat{\tilde{\phi}}_1 - \hat{\phi}_1$

Figure 3.9: Comparison of VESPA(5) and ESPRIT(8) in the presence of random perturbations on $\{g_i\}_{i=2}^{8}$ and SNR = 0 dB: (a) sample mean of $\hat{\tilde{\phi}}_1 - \phi_1$; (b) sample STD of $\hat{\tilde{\phi}}_1 - \phi_1$.

# Chapter 4

# A Gradient-Based Target Tracking Method Using Cumulants

A complete batch realization of ESPRIT [67] needs at least $\mathcal{O}(M^3)$ operations, where $M$ is the number of sensors, to perform a complete SVD or an eigenvalue decomposition (EVD). By "complete" we mean that both the singular vectors and singular values, or, eigenvectors and eigenvalues, are computed. This complexity is too costly for real-time processing.

In real-time processing, adaptive implementation is required to reduce the computational complexity, and reflect the time-varying features of the signals. For subspace methods, the adaptive implementation of second-order-statistics-based algorithms start with applying SVD [48, 49, 65, 100] or QR-decomposition [16] to the data matrix (e.g., Fig. 4.1), and they achieve high computational efficiency such as $\mathcal{O}(MP^2)$ or even $\mathcal{O}(MP)$, for an M-sensor array and P-source scenario.

On the other hand, cumulant-based subspace methods [5, 7, 13, 63] have been developed to extract DOAs and signal recovery with extra features, including less constraints on the array geometry and immunity to additive Gaussian noise. These batch methods are of complexity $\mathcal{O}(M^3)$. In this Chapter, we will develop a *recursive* implementation of VESPA [13] as a first attempt toward the adaptive processing of cumulant-based methods.

Consider the detailed steps of a realization of TLS-VESPA, as shown in Fig. 4.1. The first SVD computes the signal subspace $\mathbf{E}_s$ from the cumulant matrix $\mathbf{C}$. The right singular vectors, contained in $\mathbf{V}_1$, are not utilized by further processing, and the singular values, contained in $\mathbf{S}_1$, are used to determine the number $P$ of sources.

The problem of tracking $\mathbf{E}_s$ is called *signal subspace tracking*, and the problem of tracking $\mathbf{E}_n$ is called *noise subspace tracking*. The complexity to perform a complete SVD of a $2M \times M$ matrix $\mathbf{C}$ is $\mathcal{O}(M^3)$. The second SVD on the $M \times 2P$ matrix $\mathbf{E}_{12}$ is of complexity $\mathcal{O}(M^2P)$ if complete SVD is performed, and the left singular vectors, contained in $\mathbf{U}_2$, is redundant. The last EVD of a $P \times P$ matrix $\Psi$ is of complexity $\mathcal{O}(P^3)$. Normally, the number of signals is smaller than the number of sensors, i.e., $P < M$; hence, the first SVD occupies most of the computation load. Our goal is to develop a recursive SVD that only tracks the required singular vectors of $\mathbf{C}$ and has complexity less than $\mathcal{O}(M^3)$.

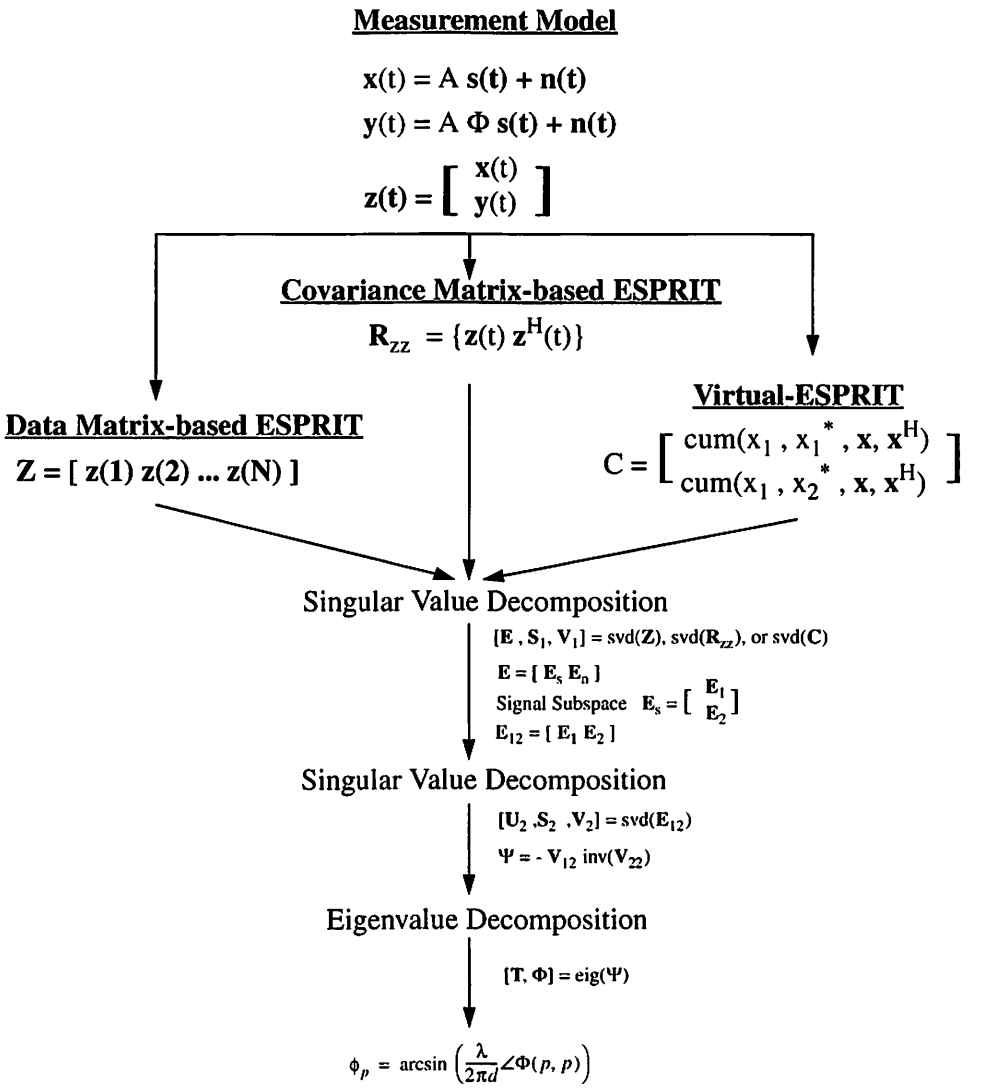



Figure 4.1: Comparisons of two versions of ESPRIT and VESPA.

Section 4.1 describes the problem of interest. Section 4.2 introduces the motivations leading to a gradient-based method, and the computation counts necessary to implement this gradient-based algorithm. Section 4.3 contains a brief stability study of this proposed method. Computer simulations of the proposed method are provided in Section 4.4. Conclusions are presented in Section 4.5.

## 4.1 Problem Description

Consider a *slowly time-varying* scenario where $P$ far-field narrowband independent sources $\{s_p(t)\}_{p=1}^{P}$, moving along directions $\{\phi_p(t)\}_{p=1}^{P}$, impinge on an M-element arbitrary array. The complex envelope representation for signal $r_i(t)$ received by the $i$-th sensor $r_i$ is modeled as

$$r_i(t) = \sum_{p=1}^{P} A_{ip}(t)\, s_p(t) + n_i(t), \quad i = 1, \cdots, M \tag{4.1}$$

where $A_{ip}(t)$ is the time-varying sensor response of $r_i$ with respect to source $s_p(t)$, and $n_i(t)$ is the Gaussian measurement noise. The sensor response $A_{ip}(t)$ is modeled as

$$A_{ip}(t) \triangleq g_i\, e^{-j2\pi[r_{ix}\sin\phi_p(t) + r_{iy}\cos\phi_p(t)]}, \quad i = 1, \cdots, M \text{ and } p = 1, \cdots, P \tag{4.2}$$

where $g_i$, a *complex number*, is the sensor gain, and, $r_{ix}$ and $r_{iy}$ (in wavelengths) are the spatial locations of sensor $r_i$ along $x$- and $y$-axes, respectively. Note that in (4.1) and (4.2), we have assumed that only azimuth angle is considered and that a planar array is used. Substituting $\boldsymbol{\kappa}_p^H = [\sin\phi_p(t)\ \cos\phi_p(t)]$ and $\mathbf{r}_i = [r_{ix}\ r_{iy}]^T$ into (1.5), we have (4.1).

Without loss of generality, we make the following assumptions:

**[AS1]** All sensor gains are unity, i.e., $g_i = 1$, $\forall i$;

**[AS2]** The first sensor $r_1$ is chosen to be the reference sensor located at the origin of the coordinate system, i.e., $r_{1x} = r_{1y} = 0$;

**[AS3]** The $x$-axis is chosen to coincide with the displacement vector connecting guiding sensors $r_1$ and $r_2$; hence, $r_{2y} = 0$;

**[AS4]** The broadside direction ($\phi = 0°$) is defined to be the direction normal to the $x$-axis;

**[AS5]** Signals are independent with nonzero fourth-order cumulants, i.e.,

$$cum(s_k(t), s_l^*(t), s_m(t), s_n^*(t)) = \begin{cases} \gamma_{4,s_k}(t), & \text{if } k = l = m = n \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Collecting signals from array outputs, we have the vector representation for the received signal model:

$$\mathbf{r}(t) = \mathbf{A}(t)\,\mathbf{s}(t) + \mathbf{n}(t) \tag{4.4}$$

which is the same as (1.6), but steering matrix $\mathbf{A}(t)$, which is defined by (4.2), is now time-varying.

Actually, assuming that the dimension of $\mathbf{A}(t)$ is fixed, subspace tracking only deals with tracking the time-varying subspace spanned by $\mathbf{A}(t)$. Specific applications of subspace tracking include adaptive beamforming, signal recovery, target tracking, and all aspects of beamspace operations. Since most performance evaluations of subspace tracking are in terms of target tracking, we define the problem of interest to be tracking the time-varying directions $\{\phi_p(t)\}_{p=1}^P$. Note that in this chapter, we assume that the number of signals is known and unchanged. The method developed in this chapter can only do target tracking. The joint rank and target tracking will be studied in Chapter 5.

## 4.2 Proposed Gradient-based Adaptive VESPA

### 4.2.1 An Optimization Problem

We introduce a constrained optimization problem, and show that the solution to it is $\mathbf{E}_s$, which spans the signal subspace (see Fig. 4.1); therefore, we only solve the first SVD of Fig. 4.1 in this subsection.

Consider the two $M \times M$ *instantaneous* cumulant matrices,

$$\mathbf{C}_1 \triangleq cum(r_1(t), r_1^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}(t)\Lambda(t)\mathbf{A}^H(t) \tag{4.5a}$$

$$\mathbf{C}_2 \triangleq cum(r_1(t), r_2^*(t), \mathbf{r}(t), \mathbf{r}^H(t)) = \mathbf{A}(t)\Phi(t)\Lambda(t)\mathbf{A}^H(t) \tag{4.5b}$$

where [29]

$$\Phi(t) \triangleq \mathrm{diag}\{e^{j[2\pi(r_{2x}\sin\phi_1(t)+r_{2y}\cos\phi_1(t))]}, e^{j[2\pi(r_{2x}\sin\phi_2(t)+r_{2y}\cos\phi_2(t))]}, \cdots, e^{j[2\pi(r_{2x}\sin\phi_P(t)+r_{2y}\cos\phi_P(t))]}\} \tag{4.6}$$

and

$$\Lambda(t) \triangleq \mathrm{diag}\{\gamma_{4,s_1}(t), \gamma_{4,s_2}(t), \cdots, \gamma_{4,s_P}(t)\}. \tag{4.7}$$

The equalities in (4.5)-(4.7) follow from Section 3.1.3 and the assumptions in Section 4.1. Note that $g_2 = 1$ (by **[AS1]**) was used to obtain (4.5)-(4.7).

To be associated with VESPA, a $2M \times M$ matrix $\mathbf{C}$ is defined, from (4.5), as

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}(t) \\ \mathbf{A}(t)\Phi(t) \end{bmatrix} \Lambda(t)\mathbf{A}(t) \tag{4.8}$$

Let us consider the following constrained optimization problem [81, 97] with respect to a $2M \times P$ matrix $\mathbf{W}$:

$$\text{maximize} \quad J = \mathrm{tr}(\mathbf{W}^H\mathbf{C}\mathbf{C}^H\mathbf{W}) \qquad \text{subject to} \quad \mathbf{W}^H\mathbf{W} = \mathbf{I} \tag{4.9}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operation. The matrix $\mathbf{C}$ corresponds to the cumulant matrix in Fig. 4.1, and $\mathbf{W}$ corresponds to the matrix $\mathbf{E}_s$ that spans the signal subspace. The orthonormalized vectors contained in $\mathbf{E}_s$ motivate us to use the constraint $\mathbf{W}^H\mathbf{W} = \mathbf{I}$. The *trace* operation is necessary since $\mathbf{W}^H\mathbf{C}\mathbf{C}^H\mathbf{W}$ is a $P \times P$ matrix. Generally, a cumulant-based matrix is not positive semi-definite; but, the Hermitian symmetric matrix $\mathbf{C}\mathbf{C}^H$ is positive semi-definite and is suitable

for the formulation of an optimization problem. We shall show that the solution, $\mathbf{W}_{max}$, to the optimization problem in (4.9) is

$$\mathbf{W}_{max} \triangleq [\mathbf{w}_{1,max}, \cdots, \mathbf{w}_{P,max}] = [\mathbf{e}_1, \cdots, \mathbf{e}_P] \triangleq \mathbf{E}_s \tag{4.10}$$

Following the notation in Fig. 4.1, let the SVD of the $2M \times M$ rectangular matrix $\mathbf{C}$ be

$$\mathbf{C} = \mathbf{E}\mathbf{S}_1\mathbf{V}_1^H \tag{4.11}$$

where

$$\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_{2M}] \tag{4.12}$$

$$\mathbf{V}_1 = [\mathbf{v}_1, \cdots, \mathbf{v}_M] \tag{4.13}$$

$$\mathbf{S}_1 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_M \\ \vdots & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \tag{4.14}$$

and the singular values are in decreasing order, i.e.,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_P \geq \sigma_{P+1} \geq \sigma_{P+2} \geq \cdots \geq \sigma_M \tag{4.15}$$

Normally, $\mathbf{C}$ is of rank $P$ [see (4.8)], and $\sigma_{P+1} = \sigma_{P+2} = \cdots = \sigma_M = 0$ under the assumption that the additive noise is Gaussian. Using (4.11), we have the SVD of $\mathbf{C}\mathbf{C}^H$ as

$$\mathbf{C}\mathbf{C}^H = \mathbf{E}\mathbf{S}_1\mathbf{V}_1^H\mathbf{V}_1\mathbf{S}_1^H\mathbf{E}^H = \mathbf{E}\mathbf{S}_1\mathbf{S}_1^H\mathbf{E}^H \triangleq \mathbf{E}\Upsilon\mathbf{E}^H \tag{4.16}$$

where

$$\Upsilon = \text{diag}\{\sigma_1^2, \cdots, \sigma_P^2, 0, \cdots, 0\} \tag{4.17}$$

Note that $\mathbf{E}$ in (4.12) is of size $2M \times 2M$.

Substituting (4.16) into (4.9), and defining

$$\mathbf{Q} \triangleq \mathbf{E}^H \mathbf{W} = [\mathbf{q}_1, \cdots, \mathbf{q}_P], \tag{4.18}$$

we have the optimization problem:

$$\text{maximize} \quad J = \text{tr}(\mathbf{Q}^H \Upsilon \mathbf{Q}) = \sum_{i=1}^{P} \sigma_i^2 \mathbf{q}_i^H \mathbf{q}_i \qquad \text{subject to} \quad \mathbf{Q}^H \mathbf{Q} = \mathbf{I} \tag{4.19}$$

Defining $J_i \triangleq \sigma_i^2 \mathbf{q}_i^H \mathbf{q}_i$ and $J \triangleq \sum_{i=1}^{P} J_i$, we have an equivalent optimization problem:

$$\text{maximize} \quad J_i = \sigma_i^2 \mathbf{q}_i^H \mathbf{q}_i, \qquad i = 1, \cdots, P \tag{4.20}$$

subject to

$$\mathbf{q}_i^H \mathbf{q}_i = 1, \qquad \mathbf{q}_i^H \mathbf{q}_j = 0, \qquad j = 1, \cdots, i-1. \tag{4.21}$$

To maximize $J_1$, we need $\mathbf{q}_{1,max} = [1, 0, 0, \cdots, 0]^T$ because $\sigma_1$ is the largest of all $\sigma_i$'s and $\mathbf{q}_{1,max}^H \mathbf{q}_{1,max} = 1$. Actually for any complex scalar $\alpha$ such that $|\alpha| = 1$, $\alpha \mathbf{q}_{1,max}$ is a solution; but for simplicity, we set $\alpha = 1$. Substituting $\mathbf{q}_{1,max}$ into (4.18), we have

$$\mathbf{w}_{1,max} \triangleq \mathbf{E}\, \mathbf{q}_{1,max} = \mathbf{e}_1 \tag{4.22}$$

To maximize $J_2$, we need $\mathbf{q}_{2,max} = [0, 1, 0, \cdots, 0]^T$ because $\mathbf{q}_2$ must satisfy the constraints in (4.21) and $\sigma_2$ is the second largest of the $\sigma_i$'s. Substituting $\mathbf{q}_{2,max}$ into (4.18), we have $\mathbf{w}_{2,max} = \mathbf{e}_2$. Continuing in this manner, we collect all the optimal $\{\mathbf{w}_{i,max}\}_{i=1}^{P}$, and have the solution shown in (4.10). This completes the derivation of (4.10).

### 4.2.2 Proposed Gradient-based Method

The constrained optimization problem introduced in the preceding subsection can be implemented through a constrained gradient-based procedure, in which weight matrix $\mathbf{W}$ is updated according to

$$\mathbf{W}(t_{k+1}) = \mathbf{W}(t_k) + \mu \, \nabla_W J \tag{4.23}$$

where the step-size parameter $\mu$ is a positive constant. The gradient of the cost function $J$ is

$$\nabla_W J = \frac{\partial}{\partial \mathbf{W}}[\mathrm{tr}(\mathbf{W}^H \mathbf{C}\mathbf{C}^H \mathbf{W})] = 2\mathbf{C}\mathbf{C}^H\mathbf{W} \tag{4.24}$$

In practice, we use the estimated value for $\mathbf{C}$, and (4.23) becomes

$$\mathbf{W}(t_{k+1}) = \mathbf{W}(t_k) + \mu \, \widehat{\mathbf{C}}(t_k)\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k) \tag{4.25}$$

where $\widehat{\mathbf{C}}(t_k)$ is the estimated $\mathbf{C}$ at time $t_k$, and the factor of 2 is absorbed into $\mu$. The updated term $\mathbf{W}(t_{k+1})$, on the left-hand-side (LHS) of (4.25), is not orthonormalized. A Gram-Schmidt Orthogonalization (GSO) procedure is then applied to the resulting $\mathbf{W}(t_{k+1})$ to keep it orthonormalized.

The complete steps for our proposed algorithm to update $\mathbf{W}(t_{k+1})$ are listed in Table 4.1. There are three major steps: update the cumulant matrix $\widehat{\mathbf{C}}(t_k)$, adjust the desired weight matrix $\mathbf{W}(t_{k+1})$, and orthonormalize the adjusted $\mathbf{W}(t_{k+1})$.

To obtain the update formulas for $\widehat{\mathbf{C}}(t_k)$, we estimate $\mathbf{C}$ using $N$ data snapshots $\{\mathbf{r}(t_i)\}_{i=k-N+1}^{k}$, assuming that $\mathbf{r}(t)$ is zero-mean:

$$\begin{aligned} \widehat{\mathbf{C}}_1(t_k) &\triangleq \mathbf{M}_1(t_k) - c_{11}(t_k)\mathbf{C}_{12}(t_k) - \mathbf{c}_{13}(t_k)\mathbf{c}_{14}^H(t_k) - \mathbf{c}_{15}(t_k)\mathbf{c}_{16}^H(t_k) \\ \widehat{\mathbf{C}}_2(t_k) &\triangleq \mathbf{M}_2(t_k) - c_{21}(t_k)\mathbf{C}_{22}(t_k) - \mathbf{c}_{23}(t_k)\mathbf{c}_{24}^H(t_k) - \mathbf{c}_{25}(t_k)\mathbf{c}_{26}^H(t_k) \end{aligned} \tag{4.26}$$

where

$$
\begin{aligned}
\mathbf{M}_1(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)r_1^*(t_i)\mathbf{r}(t_i)\mathbf{r}^H(t_i) \\
c_{11}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)r_1^*(t_i) \\
\mathbf{C}_{12}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} \mathbf{r}(t_i)\mathbf{r}^H(t_i) \\
\mathbf{c}_{13}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{14}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{15}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1^*(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{16}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1^*(t_i)\mathbf{r}(t_i) \\
\mathbf{M}_2(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)r_2^*(t_i)\mathbf{r}(t_i)\mathbf{r}^H(t_i) \\
c_{21}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)r_2^*(t_i) \\
\mathbf{C}_{22}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} \mathbf{r}(t_i)\mathbf{r}^H(t_i) \\
\mathbf{c}_{23}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{24}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_2(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{25}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_2^*(t_i)\mathbf{r}(t_i) \\
\mathbf{c}_{26}(t_k) &\triangleq \tfrac{1}{N}\textstyle\sum_{i=k-N+1}^{k} r_1^*(t_i)\mathbf{r}(t_i)
\end{aligned}
\tag{4.27}
$$

and, $r_1(t_i)$ and $r_2(t_i)$ are the 1st and 2nd entry of $\mathbf{r}(t_i)$, respectively. Some properties can be used to reduce the computations in (4.26)-(4.27). The $M \times M$ matrix $\mathbf{C}_{12}(t_k)$ contains $c_{11}(t_k)$, $\mathbf{c}_{15}(t_k)$, $\mathbf{c}_{16}(t_k)$, $c_{21}(t_k)$, $\mathbf{c}_{25}(t_k)$, and $\mathbf{c}_{26}(t_k)$. There are identical terms such as $\mathbf{C}_{12}(t_k)$ and $\mathbf{C}_{22}(t_k)$, and, $\mathbf{c}_{13}(t_k)$ and $\mathbf{c}_{23}(t_k)$. $\mathbf{c}_{13}(t_k)$ is identical to $\mathbf{c}_{14}(t_k)$, and $\mathbf{c}_{15}(t_k)$ is identical to $\mathbf{c}_{16}(t_k)$.

To update $\mathbf{C}_{12}(t_k)$ (see Table 4.1), we need $M^2 + 3M$ (complex) multiplications, including $M$ multiplications for $\frac{1}{\sqrt{N}}\mathbf{r}(t_k)$ and $\frac{1}{2}M^2 + \frac{1}{2}M$, rather than $M^2$, for $[\frac{1}{\sqrt{N}}\mathbf{r}(t_k)][\frac{1}{\sqrt{N}}\mathbf{r}(t_k)]^H$ because of conjugate symmetry, and, the same number of multiplications for $[\frac{1}{\sqrt{N}}\mathbf{r}(t_{k-N+1})][\frac{1}{\sqrt{N}}\mathbf{r}(t_{k-N+1})]^H$. The complexity for updating $\mathbf{c}_{13}$ is $2M+2$; but for simplicity, the zero-order term 2 is dropped to give $2M$. The complexity to update $\widehat{\mathbf{C}}_1(t_k)$ is $\frac{3}{2}M^2 + \frac{3}{2}M$, including $\frac{1}{2}M^2 + \frac{1}{2}M$ multiplications each for $c_{11}(t_k)\mathbf{C}_{12}(t_k)$, $\mathbf{c}_{13}(t_k)\mathbf{c}_{13}^H(t_k)$, and $\mathbf{c}_{16}(t_k)\mathbf{c}_{16}^H(t_k)$. The computation needed for $\widehat{\mathbf{C}}_2(t_k)$ is higher than that for $\widehat{\mathbf{C}}_1(t_k)$ due to non-conjugate symmetry of $\widehat{\mathbf{C}}_2(t_k)$. Adding up the overall complexities, we have the computation complexity for updating $\widehat{\mathbf{C}}(t_k)$ equal to $8M^2 + 14M$.

The complexity to adjust $\mathbf{W}(t_{k+1})$ is $4M^2P + 2MP$ multiplications, including: the complexity for $\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k)$, which is $2M^2P$ multiplications, the complexity for $\widehat{\mathbf{C}}(t_k)[\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k)]$, which is $2M^2P$, and the complexity for $\mu[\widehat{\mathbf{C}}(t_k)\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k)]$ which is $2MP$. Finally, the GSO procedure for a $2M \times P$ matrix $\mathbf{W}(t_{k+1})$ requires $MP^2 + 3MP$ multiplications [97].

Table 4.1: Summary of the proposed LMS-type VESPA.

| **Update** | **Complexity** |
|---|---|
| % Initialization | |
| $\mathbf{W}(0) = [\mathbf{I}_{P\times P}; \mathbf{0}_{(2M-P)\times P}]^T$ | |
| % Update $\widehat{\mathbf{C}}(t_k)_{2M\times M}$ | $\underline{8M^2 + 14M}$ |
| $\mathbf{C}_{12}(t_k)_{M\times M} = \mathbf{C}_{12}(t_{k-1}) + \frac{1}{\sqrt{N}}\mathbf{r}(t_k)[\frac{1}{\sqrt{N}}\mathbf{r}(t_k)]^H$ | $M^2 + 3M$ |
| $\quad -\frac{1}{\sqrt{N}}\mathbf{r}(t_{k-N+1})[\frac{1}{\sqrt{N}}\mathbf{r}(t_{k-N+1})]^H$ | |
| $c_{11}(t_k)_{1\times 1}$ = (1,1)-th entry of $\mathbf{C}_{12}(t_k)$ | |
| $\mathbf{c}_{13}(t_k)_{M\times 1} = \mathbf{c}_{13}(t_{k-1}) + \frac{1}{N}r_1(t_k)\mathbf{r}(t_k) - \frac{1}{N}r_1(t_{k-N+1})\mathbf{r}(t_{k-N+1})$ | $2M$ |
| $\mathbf{c}_{16}(t_k)_{M\times 1}$ = conjugate transpose of the 1st row of $\mathbf{c}_{12}(t_k)$ | |
| $\mathbf{M}_1(t_k)_{M\times M} = \mathbf{M}_1(t_{k-1}) + \frac{1}{\sqrt{N}}r_1^*(t_k)\mathbf{r}(t_k)[\frac{1}{\sqrt{N}}r_1^*(t_k)\mathbf{r}(t_k)]^H$ | $M^2 + 3M$ |
| $-[\frac{1}{\sqrt{N}}r_1^*(t_{k-N+1})\mathbf{r}(t_{k-N+1})][\frac{1}{\sqrt{N}}r_1^*(t_{k-N+1})\mathbf{r}(t_{k-N+1})]^H$ | |
| $\widehat{\mathbf{C}}_1(t_k)_{M\times M} = \mathbf{M}_1(t_k) - c_{11}(t_k)\mathbf{C}_{12}(t_k) - \mathbf{c}_{13}(t_k)\mathbf{c}_{13}^H(t_k)$ | $\frac{3}{2}M^2 + \frac{3}{2}M$ |
| $\quad -\mathbf{c}_{16}(t_k)\mathbf{c}_{16}^H(t_k)$ | |
| $\mathbf{M}_2(t_k)_{M\times M} = \mathbf{M}_2(t_{k-1}) + [\frac{1}{N}r_1(t_k)\mathbf{r}(t_k)][r_2^*(t_k)\mathbf{r}^H(t_k)]$ | $2M^2 + 2M$ |
| $\quad -[\frac{1}{N}r_1(t_{k-N+1})\mathbf{r}(t_{k-N+1})][r_2^*(t_{k-N+1})\mathbf{r}^H(t_{k-N+1})]$ | |
| $c_{21}(t_k)_{1\times 1}$ = (1,2)-th entry of $\mathbf{C}_{12}(t_k)$ | |
| $\mathbf{c}_{24}(t_k)_{1\times M} = \mathbf{c}_{24}(t_{k-1}) + \frac{1}{N}r_2(t_k)\mathbf{r}(t_k) - \frac{1}{N}r_2(t_{k-N+1})\mathbf{r}(t_{k-N+1})$ | $2M$ |
| $\mathbf{c}_{25}(t_k)_{M\times 1}$ = 2nd column of $\mathbf{C}_{12}(t_k)$ | |
| $\widehat{\mathbf{C}}_2(t_k)_{M\times M} = \mathbf{M}_2(t_k) - c_{21}(t_k)\mathbf{C}_{12}(t_k) - \mathbf{c}_{13}(t_k)\mathbf{c}_{24}^H(t_k)$ | $\frac{5}{2}M^2 + \frac{1}{2}M$ |
| $\quad -\mathbf{c}_{25}(t_k)\mathbf{c}_{16}^H(t_k)$ | |
| $\widehat{\mathbf{C}}(t_k) = [\widehat{\mathbf{C}}_1^H(t_k) \mid \widehat{\mathbf{C}}_2^H(t_k)]^H$ | |
| % Adjust $\mathbf{W}(t_{k+1})_{2M\times P}$ | |
| $\mathbf{W}(t_{k+1}) = \mathbf{W}(t_k) + \mu\, \widehat{\mathbf{C}}(t_k)\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k)$ | $\underline{4M^2P + 2MP}$ |
| % Gram-Schmidt Orthogonalization Procedure | $\underline{MP^2 + 3MP}$ |
| % applied to $\mathbf{W}(t_{k+1})$ | |
| % Next Step $t_k = t_{k+1}$ | |
| % Total complexity | $4M^2P + MP^2$ |

A complete iteration of all the steps in Table 4.1 requires complexity of

$$4M^2P + MP^2 + 8M^2 + 5MP + 14M \tag{4.28}$$

Hence, the computations for Table 4.1 is of complexity $\mathcal{O}(M^2P)$.

Table 4.1 only computes the first SVD of Fig. 4.1. For each iteration, our gradient-based VESPA for target tracking computes the first SVD according to Table 4.1, and performs the second SVD and last EVD non-adaptively. Note that the second SVD and last EVD are of complexity $\mathcal{O}(M^2P)$ and $\mathcal{O}(P^3)$, respectively. We refer to our target tracking method, which is of complexity $\mathcal{O}(M^2P)$, as least-mean-squared-VESPA (LMS-VESPA), because it is a stochastic gradient-based method.

## 4.3 Stability Study

The analysis of convergence rates and stability of the adaptive eigensubspace methods can be accomplished by following the classical procedures that assume independence of data and weight variations, as in [35].

The adaptation formula for the signal subspace estimator $\mathbf{W}(t_{k+1})$ is,

$$\mathbf{W}(t_{k+1}) = \mathbf{W}(t_k) + \mu\, \widehat{\mathbf{C}}(t_k)\widehat{\mathbf{C}}^H(t_k)\mathbf{W}(t_k) \tag{4.29a}$$

$$\text{GSO of the columns of } \mathbf{W}(t_{k+1}) \tag{4.29b}$$

For stationary data, the mean of $\mathbf{W}(t_{k+1})$ is

$$E[\mathbf{W}(t_{k+1})] = (\mathbf{I} + \mu\mathbf{C}\mathbf{C}^H)\, E[\mathbf{W}(t_k)] \tag{4.30a}$$

$$E[\mathbf{W}(t_{k+1})] = \text{GSO of the columns of } E[\mathbf{W}(t_{k+1})] \tag{4.30b}$$

where independence among $\widehat{\mathbf{C}}(t_k)$, $\widehat{\mathbf{C}}^H(t_k)$, and $\mathbf{W}(t_k)$, and, unbiasedness of $\widehat{\mathbf{C}}(t_k)$ are assumed. The assumption of independence is widely used in analyzing LMS-type algorithms [35]. The GSO of the columns of $E[\mathbf{W}(t_{k+1})]$ is equivalent to the unitary Q-matrix of the QR-decomposition of $E[\mathbf{W}(t_{k+1})]$ [23, p. 136]. Let $\mathbf{B} = \mathbf{I} + \mu\mathbf{C}\mathbf{C}^H$, then the dynamics of the mean of the eigensubspace estimators can be treated as an

orthogonal iteration [23, p. 211] which is the generalization of the power method, and is stated as follows:

$$E[\mathbf{W}(t_{k+1})] = \mathbf{B}\, E[\mathbf{W}(t_k)] \tag{4.31a}$$

$$E[\mathbf{W}(t_{k+1})] = \text{Q-matrix of the QR-decomposition of } E[\mathbf{W}(t_{k+1})] \tag{4.31b}$$

It has been shown in [23, pp. 211-212] that the convergence rate of the $P$-dominant subspace, which is the span of eigenvectors corresponding to the $P$ largest eigenvalues of $\mathbf{B}$, is proportional to the $k$th power of the convergence index $\rho$,

$$\rho = \left| \frac{(P+1) - \text{th largest eigenvalue of } \mathbf{B}}{P - \text{th largest eigenvalue } \mathbf{B}} \right| \tag{4.32}$$

Thus, the stationary convergence characteristics of $\mathbf{W}(t_k)$ will only depend on the eigenvalues of $\mathbf{B}$. Here, we utilize this idea to analyze the convergence rates of $\mathbf{W}(t_k)$.

The condition for stability (convergence in the mean) is given [97] by

$$0 < \mu < \infty \tag{4.33}$$

Using (4.17)-(4.18), the convergence index $\rho$ of (4.32) becomes

$$\rho = \frac{1 + \mu\sigma_P^2}{1 + \mu\sigma_{P+1}^2} = 1 + \mu\sigma_P^2 \tag{4.34}$$

where $\sigma_P$ is the smallest non-zero singular value of the rank-P cumulant matrix $\mathbf{C}$. One important implication of (4.34) is that LMS-VESPA with large $\mu$ has a faster convergence rate.

## 4.4 Experiments

### 4.4.1 Effect of Window Size

In this experiment, the familiar VESPA(5) configuration (see Fig. 3.2) was used with LMS-VESPA; the complete implementation is denoted LMS-VESPA(5). Our

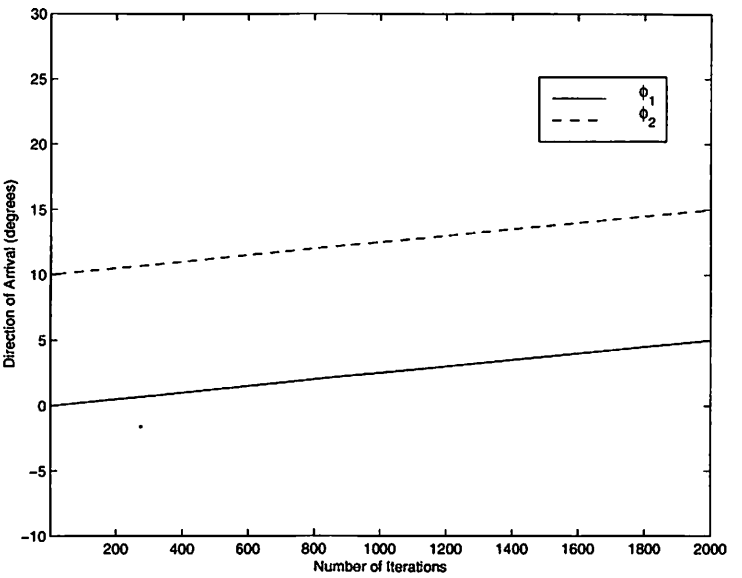



Figure 4.2: Two moving targets.

objective is to study how the window size, $N$, affects the performance of LMS-VESPA(5).

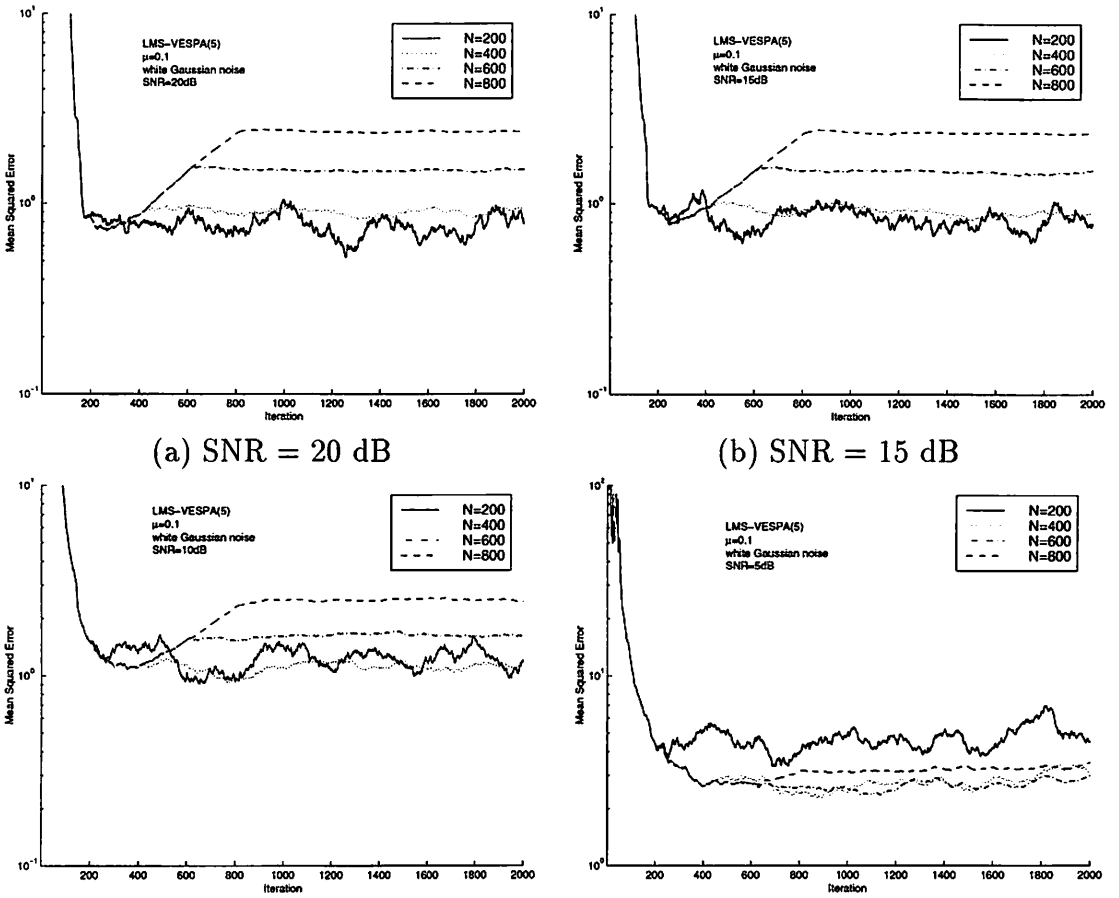
Each measurement was contaminated with additive Gaussian noise that was both spatially and temporally white, and was independent of the signals. Two phase modulated (PM) signals were moving as shown in Fig. 4.2. We assumed that the two signals had the same power; therefore, one SNR is enough to describe the SNR situation. The weight vector was initialized to be $\mathbf{W}(0) = [\mathbf{I}_{P\times P}; \mathbf{0}_{2M-P)\times P}]^T$. If the iteration $k$ is less than the window size $N$, the cumulants are computed as an average over the available $k$ snapshots.

We ran 100 MC simulations of LMS-VESPA(5) using a fixed $\mu = 0.1$ and different window sizes, N = 200, 400, 600, and 800, for SNR = 20, 15, 10, 5 dB. The resulting MSEs,

$$\begin{aligned} \text{MSE}(t_k) &= \frac{1}{2}E\{|\hat{\phi}_1(t_k) - \phi_1(t_k)|^2\} \\ &\quad + \frac{1}{2}E\{|\hat{\phi}_2(t_k) - \phi_2(t_k)|^2\}, \quad k = 1, 2, \cdots \end{aligned} \tag{4.35}$$

versus the number of iterations are plotted in Fig. 4.3.

In Fig. 4.3(a), LMS-VESPA(5) using $N = 200$ outperforms all the other cases of LMS-VESPA(5), because the short windowing causes LMS-VESPA(5) to respond




(a) SNR = 20 dB (b) SNR = 15 dB

(c) SNR = 10 dB (d) SNR = 5 dB

Figure 4.3: Mean squared errors of target tracking by LMS-VESPA(5) using different window sizes versus the number of iterations.

quickly to the variation in the signals, i.e., the shorter the window, the smaller the MSE when SNR is high. LMS-VESPA(5) with $N = 200$ degrades as SNR decreases [see Figs. 4.3(b) and (c)], and for SNR = 5 dB, it performs very poorly as shown in Fig. 4.3(d). This is due to the fact the cumulant-based methods need longer data length to accurately estimate the desired quantities.

We conclude from this experiment that if the signals move quickly, short window size should be selected for LMS-VESPA, and that if the SNR is low, long window size should be selected.

### 4.4.2 Effect of Step-Size Parameter

In this experiment, we study how the step-size parameter $\mu$ affects the performance of LMS-VESPA(5). A similar experiment to that in Section 4.4.1 was performed; but, we varied $\mu = 0.01, 0.1$, and 1, and, fixed $N = 300$ for LMS-VESPA(5) under different SNR conditions. The MSEs, over 100 MC runs, versus the number of iterations for all the cases are plotted in Fig. 4.4.

In Figs. 4.4(a)-(c), LMS-VESPA(5) using $\mu = 1$ has the lowest MSE, indicating that LMS-VESPA using large step-size parameter responds faster to the variations in the signal directions when the SNR is moderate (10 dB) or high (20 dB). When the SNR is low (5 dB), LMS-VESPA(5) using $\mu = 0.1$ and 1 have approximately the same MSEs. In Figs. 4.4(a)-(d), LMS-VESPA(5) using $\mu = 0.01$ has the largest MSE, which results mainly from the signal variations; hence, $\mu = 0.01$ is too small for all cases.

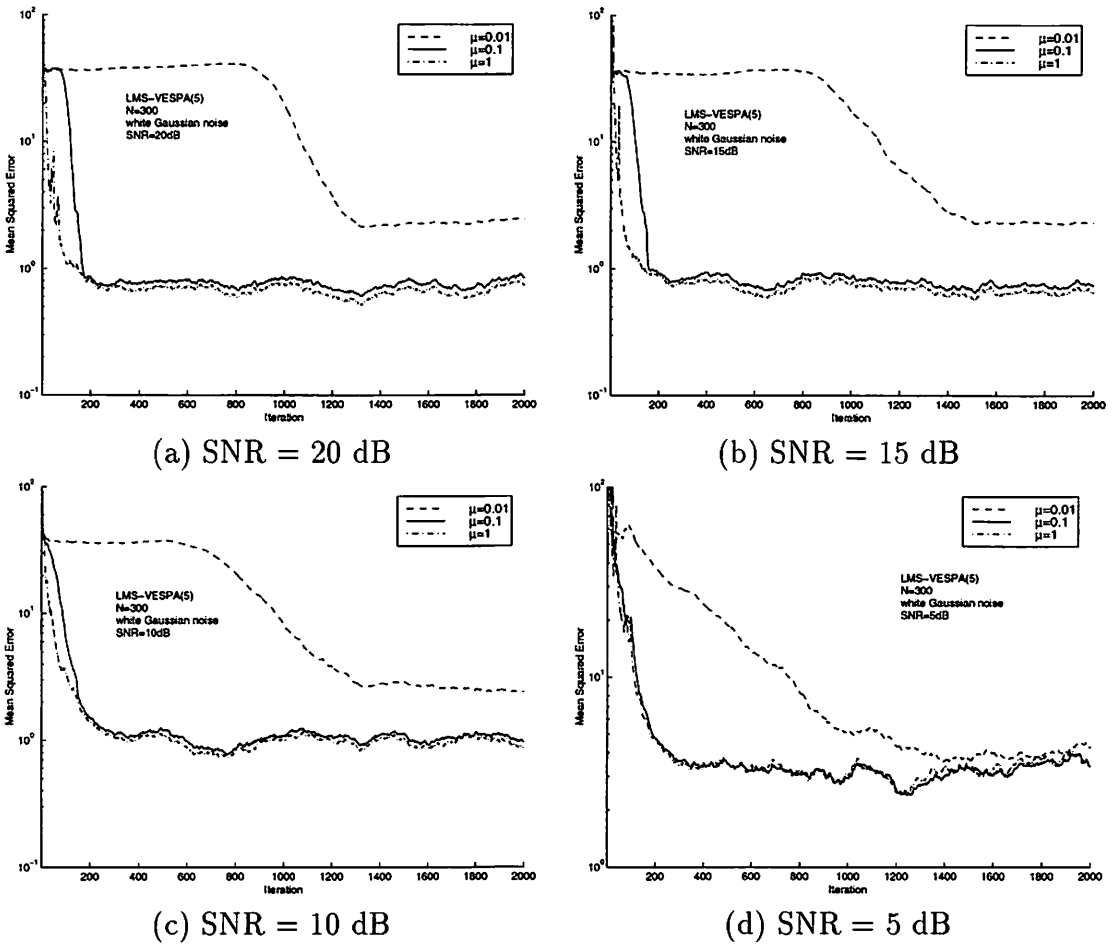
We conclude that $\mu$ should be selected large enough to respond to the variations in the signal directions.

### 4.4.3 Colored Gaussian Noise

In this experiment, we compare the tracking by LMS-VESPA(5) and block-adapted ESPRIT(8) for a colored Gaussian noise environment.

The block-adapted ESPRIT(8) uses the ESPRIT(8) configuration (see Fig. 3.2) and block-adapted ESPRIT algorithm. From Fig. 4.1, it computes the SVD of the covariance matrix, estimated from the most recent $N$ snapshots, and performs the second SVD and last EVD non-adaptively as does LMS-VESPA(5). LMS-VESPA(5) uses a 5-element array, which is contained in the 8-element array used by block-adapted ESPRIT(8). LMS-VESPA(5) and block-adapted ESPRIT(8) use the same window size N to estimate the cumulant-based matrices and covariance matrices, respectively.

In this experiment, two signals are moving as depicted in Fig. 4.5. Note that they approach each other before $t_k = 1500$, and then spread apart from each other after $t_k = 1500$. For $1 < t_k < 2000$, the scenario corresponds to closely spaced signals, and for $2001 < t_k < 4000$, the scenario corresponds to far-apart signals.




(a) SNR = 20 dB (b) SNR = 15 dB

(c) SNR = 10 dB (d) SNR = 5 dB

Figure 4.4: Mean squared errors of target tracking by LMS-VESPA(5) using different step-sizes versus the number of iterations.

Additive noise is stationary, colored, and Gaussian with zero-mean and covariance matrix $\mathbf{R}_{nn}$ [13, 14], where

$$\mathbf{R}_{nn} = \sum_{\phi=30^\circ}^{60^\circ} \frac{1}{31}\mathbf{a}(\phi)\mathbf{a}^H(\phi) \tag{4.36}$$

in which the scale factor $\frac{1}{31}$ is used to normalized the covariance matrix such that its diagonal entries are unity. It is known that covariance-matrix-based methods work very poorly in the presence of colored noise, and that cumulant-based methods are immune theoretically to such noise. The colored noise of (4.36) simulates the situation where multiple stationary interferences impinge on the array from directions

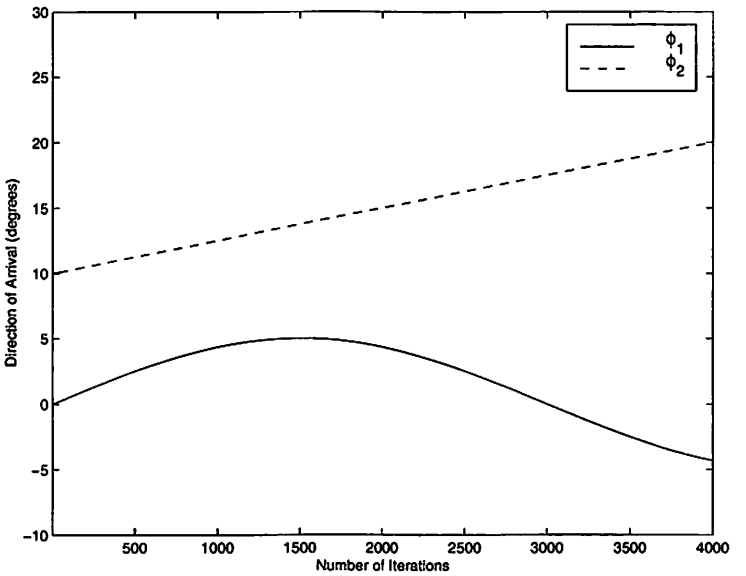


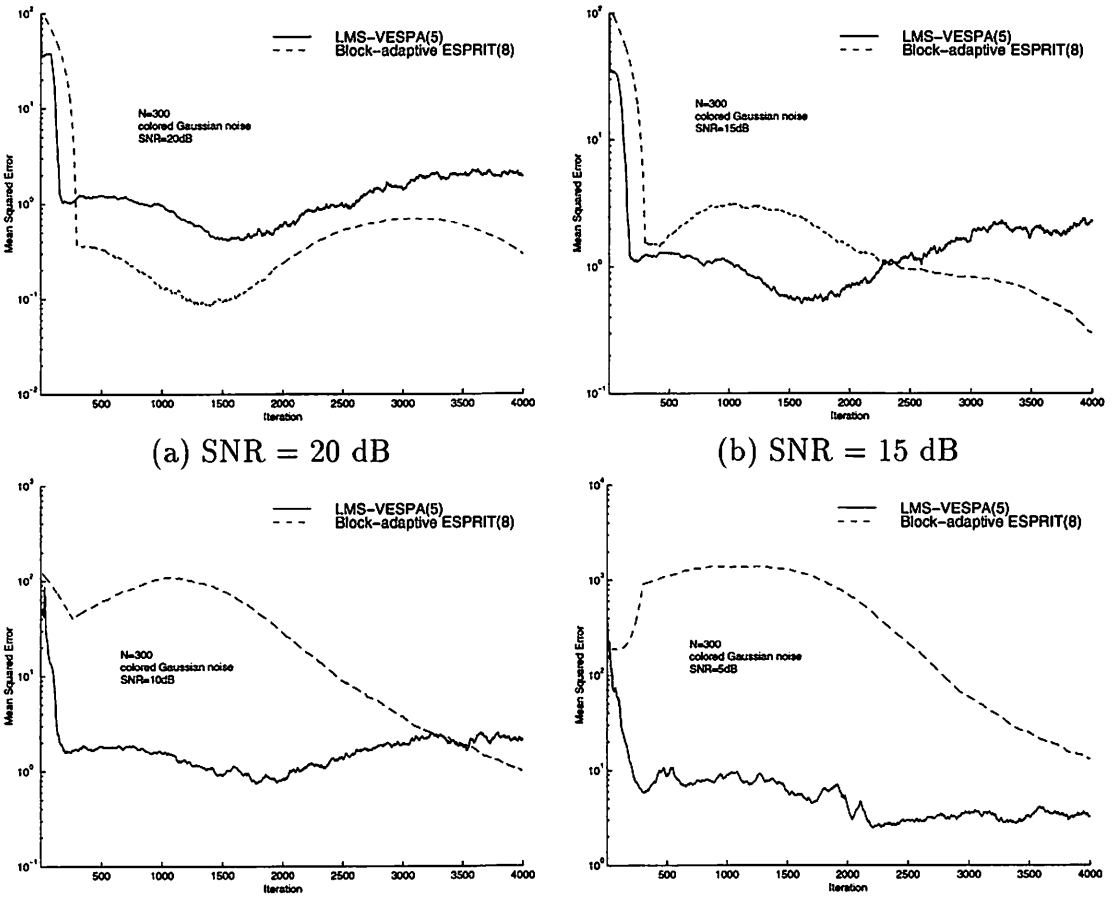

Figure 4.5: Two moving targets.

$30°, 31°, \cdots, 60°$. All signals are assumed to have the same power; and, the spatial noises have identical powers, as revealed by the identical diagonal entries of $\mathbf{R}_{nn}$. Assuming the two signals in this experiment are of the same power, we use a single SNR value to indicate the SNR condition between the multiple signals and noises.

MSEs versus the number of iterations for SNR = 20, 15, 10, and 5 dB are plotted in Fig. 4.6. In Fig. 4.6(a), the MSEs are mainly caused by the variations in the signal directions, because SNR = 20 dB. Block-adapted ESPRIT(8) has a lower MSE, because second-order statistics converge faster than HOS.

Colored noise takes effect in Figs. 4.6(b)-(d). When the noise is slight (15 dB) and the signals are close ($t_k < 2200$), LMS-VESPA(5) has a smaller MSE than block-adapted ESPRIT(8), as shown in Fig. 4.6(b); but when the signals are far-apart ($t_k > 2200$), block-adapted ESPRIT(8) has a lower MSE, revealing that this small amount of colored noise seriously deteriorates the tracking by block-adapted ESPRIT(8) when the signals are closely spaced. Similar revelation is given by Fig. 4.6(c). In Fig. 4.6(d), when the SNR = 5 dB, LMS-VESPA(5) has a lower MSE than block-adapted ESPRIT(8) no matter whether the signals are closely spaced or far-apart.

We conclude that when the signals are closely spaced, block-adapted ESPRIT(8) suffers even from slight colored noise, and that when the SNR is poor (5 dB), whether the signals are close or not, LMS-VESPA(5) is still robust to such noise.




(a) SNR = 20 dB (b) SNR = 15 dB

(c) SNR = 10 dB (d) SNR = 5 dB

Figure 4.6: Mean squared errors of target tracking by LMS-VESPA(5) and block-adapted ESPRIT(8) versus the number of iterations in the presence of colored Gaussian noise.

## 4.5 Conclusions

We proposed an LMS-type VESPA which needs complexity of $\mathcal{O}(M^2P)$. LMS-VESPA computes the first SVD of TLS-VESPA recursively, and performs the second SVD and last EVD non-adaptively. To use LMS-VESPA, the window size and step-size parameter should be selected appropriately to give good tracking results. In addition to hardware saving, when the additive noise is colored Gaussian, LMS-VESPA outperforms block-adapted ESPRIT, in terms of lower MSEs, for most of the SNR conditions.

# Chapter 5

# Cumulant-Based Subspace Tracking

In batch processing, the first step of all subspace methods computes either the signal subspace or the noise subspace. For correlation-matrix-based methods (e.g., TLS-ESPRIT [67]), EVD or SVD is applied to the sample correlation matrix of the array outputs. The number of signals is determined from the singular values of the sample correlation matrix using optimal criteria like the Akaike information criterion (AIC) [1] or minimum description length (MDL) [93]. Singular vectors that are associated with the dominant singular values form a basis for the signal subspace, and the remaining singular vectors form the basis for the noise subspace. Note that signal and noise subspaces are orthogonal complements of each other.

For data-matrix-based methods, the procedure is the same as for correlation-matrix-based methods, except that the correlation matrix is replaced by its data matrix counterpart.

In a slowly time-varying scenario, subspace tracking [16, 40, 48, 64, 97] is the problem of interest and is decomposed into two problems: tracking the number of signals and tracking the signal or noise subspaces. In the numerical linear algebra literature, these two eigenstructure update problems are named *rank tracking* (of the dominant subspace) and *subspace tracking*, respectively. For each iteration, a conventional tracking algorithm updates the eigenstructure as a new snapshot arrives. The algorithm then determines whether the rank of the signal subspace should increase (if a new signal appears), decrease (if a signal disappears), or remain unchanged (no signal appears or disappears) from the preceding update. This newly computed eigenstructure is then combined with, e.g., root-MUSIC [2] and ESPRIT [49], to perform target tracking.

Fast computation is a major goal to implement algorithms adaptively. From the computational viewpoint, eigenstructure update algorithms are classified (references are given below) as requiring $\mathcal{O}(M^2P)$, $\mathcal{O}(M^2)$, $\mathcal{O}(MP^2)$, or $\mathcal{O}(MP)$ flops per iteration, where $M$ is the number of sensors, and $P$ ($P < M$) is the dimension of the signal subspace. In this dissertation, a *flop* is defined to be a multiplication of two complex numbers. This wide range of computational complexity is due to the fact that some algorithms update the complete eigenstructure, with or without the explicit computations of the sample correlation matrix or data matrix, whereas others update only the signal or the noise subspace [8, 10]. Lanczos-type [24] and gradient-type [97] algorithms were first developed for tracking approximate eigenstructure of the correlation matrix and achieved complexity $\mathcal{O}(M^2P)$. Because SVD is difficult to update in its standard form, fast algorithms are always based on other decompositions of the data matrix. For example, Stewart's rank-revealing URV algorithm [72, 49], requiring $\mathcal{O}(M^2)$ flops per iteration, tracks the signal subspace and preserves signal eigenstructure information. Rabideau's [65] fast subspace tracking (FST) algorithm, requiring $\mathcal{O}(MP)$ flops, is a spherically averaged URV, which tracks approximate signal eigenstructure and averaged noise singular values. Degroat's [9] rank-1 signal averaged (ROSA) algorithm, requiring $\mathcal{O}(MP)$ flops, computes averaged signal and noise eigenvalues. Note that the class of $\mathcal{O}(MP)$ algorithms always operates on the data matrix, because a rank-1 update of a correlation matrix needs $M^2/2$ operations. Of the algorithms studied by Comon and Golub [8], Karasalo's [39] algorithm was judged to make the best tradeoff between performance and complexity. This algorithm achieves $\mathcal{O}(MP^2)$ complexity by averaging noise singular values, and is a maximum likelihood algorithm under a Gaussian noise assumption [8].

Applications of HOS for array signal processing were originally motivated by the fact that cumulants can remove colored Gaussian noise. Later developments showed that using cumulants saves hardware cost [13], imposes less constraints on array configurations [13, 29], and is more robust to model perturbations [50]. From the preceding paragraph, we know that all fast subspace tracking algorithms are of complexity $\mathcal{O}(MP)$ and compute the eigenstructure of the data matrix. Our objective is to develop a cumulant-based subspace tracking algorithm that is of $\mathcal{O}(MP)$ complexity.

Our proposed cumulant-based approach is one of preprocessing the actual data snapshots. In this approach, we estimate a cumulant-based vector, called the *virtual snapshot*, from the most recent data for each iteration. This virtual snapshot contains the same information as the actual measurement, and is a good substitute for the actual snapshot in all existing subspace tracking methods. Although there are many applications of subspace techniques, in this chapter, we will confine ourself to target and rank tracking, as does most of the subspace tracking literature.

The structure of this chapter is as follows. Section 5.1 describes the problem of interest. Section 5.2 introduces the concept of virtual snapshots. Section 5.3 considers practical problems encountered in implementation. Section 5.4 demonstrates how to realize VESPA for target tracking. Section 5.5 discusses some properties of our cumulant-based methods. Section 5.6 presents two simulation results of applying adaptive VESPA for target and rank tracking. Section 5.7 presents our conclusions.

## 5.1 Problem Description

In this chapter, we investigate the same problem as that in Chapter 4. The problem description of Chapter 4, Section 4.1, is used in this chapter, except that we additionally assume that the number of signals is unknown. Our problem of interest is to do both rank and target tracking. Note that LMS-VESPA of Chapter 4 can only do target tracking.

## 5.2 Cumulant-based Preprocessing

In contrast with stationary environments in [13, 29], we consider the time-varying scenario here. Let's consider the instantaneous cumulant vector
$cum(r_m(t), r_u^*(t), \mathbf{r}(t), r_v^*(t))$, whose $i$-th component is $cum(r_m(t), r_u^*(t), r_i(t), r_v^*(t))$ and $m, u, i, v \in \{1, 2, \cdots, M\}$. Because $r_i(t), \forall i$, is nonstationary,
$cum(r_m(t), r_u^*(t), r_i(t), r_v^*(t))$ is time-dependent (see [29]), i.e.,

$$cum(r_m(t), r_u^*(t), r_i(t), r_v^*(t)) = \sum_{p=1}^{P} \gamma_{4,s_p}(t) A_{mp}(t) A_{up}^*(t) A_{ip}(t) A_{vp}^*(t) \quad (5.1)$$

where, in the derivation of (5.1), assumption **[AS5]** of Chapter 4 was used and we also used the fact that Gaussian noise is removed by cumulants. For notational convenience, we let $\mathbf{c}_{muv}(t)$ denote $cum(r_m(t), r_u^*(t), \mathbf{r}(t), r_v^*(t))$, i.e.,

$$\mathbf{c}_{muv}(t) \triangleq cum(r_m(t), r_u^*(t), \mathbf{r}(t), r_v^*(t)) \tag{5.2}$$

where the arbitrarily chosen subscripts represent the three arguments in $cum(\cdot)$, i.e., $m, u, v \in \{1, 2, \cdots, M\}$. We refer to $\mathbf{c}_{muv}(t)$ as a "virtual snapshot". Substituting (5.1) into (5.2), and arranging the terms properly, we obtain

$$\mathbf{c}_{muv}(t) \;=\; \mathbf{A}(t)\, \tilde{\mathbf{s}}_{muv}(t) \tag{5.3}$$

where

$$\tilde{\mathbf{s}}_{muv}(t) \;=\; \begin{bmatrix} \gamma_{4,s_1}(t) A_{m1}(t) A_{u1}^*(t) A_{v1}^*(t) \\ \vdots \\ \gamma_{4,s_P}(t) A_{mP}(t) A_{uP}^*(t) A_{vP}^*(t) \end{bmatrix} \tag{5.4}$$

Observe that noise-free random vector $\mathbf{r}(t) = \mathbf{A}(t)\,\mathbf{s}(t)$ and deterministic $\mathbf{c}_{muv}(t) = \mathbf{A}(t)\,\tilde{\mathbf{s}}_{muv}(t)$ are both contained in the space spanned by $\mathbf{A}(t)$. Furthermore, noise-free $\mathbf{r}(t)$ and $\mathbf{c}_{muv}(t)$ are unavailable, because $\mathbf{r}(t)$ is contaminated by additive noise and $\mathbf{c}_{muv}(t)$ has to be estimated from noisy measurements. Note that the column space of $\mathbf{A}(t)$ is the signal subspace. In the following, we compare how noise-free $\mathbf{r}(t)$ and $\mathbf{c}_{muv}(t)$ provide update information for a subspace tracking algorithm, and see if we can simply replace $\mathbf{r}(t)$ by $\mathbf{c}_{muv}(t)$ in already-developed subspace tracking algorithms.

We assume that the readers are familiar with how data-matrix-based subspace tracking algorithms process new snapshots. An example of a data-matrix-based tracking algorithm is presented in Table 5.1.

At time $t_k$, $\mathbf{A}(t_k) = [\mathbf{a}_1(t_k), \cdots, \mathbf{a}_P(t_k)]$ is assumed to be of dimension $M \times P$, i.e., there are P signals present. Here, we briefly review how each new snapshot $\mathbf{r}(t_n), \forall n > k$, provides update information for the data-matrix-based subspace tracker. For purposes of this discussion, we assume that rank change may occur

Table 5.1: Rabideau's Rank Adaptive Fast Subspace Tracking [65]

| Update | Flops |
|---|---|
| % Initialization: $\mathbf{R}_s(t_0) = \mathbf{0}$, $\widehat{\sigma}_n(t_0) = 0$, $\mathbf{V}_s(t_0) = \mathbf{I}$. | |
| % As time increases, $k = 1, 2, 3, \cdots$ | |
| $\mathbf{r}_s = \mathbf{V}_s^H(t_{k-1})\,\mathbf{r}(t_k)$ | $MP$ |
| $\gamma = \|\mathbf{r}(t_k) - \mathbf{V}_s(t_{k-1})\,\mathbf{r}_s\|_2$; $\mathbf{v}_n = \frac{\mathbf{r}(t_k) - \mathbf{V}_s(t_{k-1})\,\mathbf{r}_s}{\gamma}$ | $MP$ |
| $\mathbf{R}_{ap} \triangleq \begin{bmatrix} \beta\mathbf{R}_s(t_{k-1}) & \mathbf{0} \\ \mathbf{0}^H & \beta\widehat{\sigma}_n(t_{k-1}) \\ \mathbf{r}_s^H & \gamma \end{bmatrix}$ | $P^2/2$ |
| % Apply left plane rotations, $\mathbf{Q}$, to $\mathbf{R}_{ap}$ such that $\mathbf{R}_0 = \mathbf{Q}\mathbf{R}_{ap}$ is upper triangular | $3P^2$ |
| % (Deflation) Compute $\mathbf{w}$ such that $\mathbf{b} = \mathbf{R}_0\mathbf{w}$ and $\|\mathbf{b}\|_2 \approx \widehat{\sigma}_P$, and, apply plane rotations, $\mathbf{Q}_1$ and $\mathbf{Q}_2$, to $\mathbf{R}_0$ such that $\mathbf{R}_1 = \mathbf{Q}_2\mathbf{R}_0\mathbf{Q}_1$ is upper triangular. | $6P^2$ |
| % (Refinement) Apply plane rotations, $\mathbf{Q}_{1a}$ and $\mathbf{Q}_{2a}$, such that the entries in the last column of $\mathbf{R}_3 = \mathbf{Q}_{2a}\mathbf{R}_1\mathbf{Q}_{1a}$ are small. Let $\widehat{\sigma}$ be the $(P+1, P+1)$th entry of $\mathbf{R}_3$. | $6P^2$ |
| % Update subspace $[\mathbf{V}_s(t_k)\ \mathbf{v}_n] = [\mathbf{V}_s(t_{k-1})\ \mathbf{v}_n]\,\mathbf{Q}_1\mathbf{Q}_{1a}$ | $12MP$ |
| % Rank change test: | |
| if $\frac{(M-P-1)(\beta\widehat{\sigma}_n(t_{k-1}))^2 + \|\widehat{\sigma}\|^2}{M-P} > T_{\text{inc}}$ % rank increase test | |
| $\mathbf{V}_s(t_k) = [\mathbf{V}_s(t_k)\ \mathbf{v}_n]$; $P = P + 1$ | |
| else if $\frac{(M-P-2)(\beta\widehat{\sigma}_n(t_{k-1}))^2 + \|\widehat{\sigma}\|^2 + \|\widehat{\sigma}_P\|^2}{M-P+1} < T_{\text{dec}}$ % rank decrease test | |
| % Repeat deflation and refinement on the principle $P \times P$ submatrix of $\mathbf{R}_3$ such that $\mathbf{R}_3$ becomes $\begin{bmatrix} & s' & s \\ \mathbf{R}_s(t_k) & \vdots & \vdots \\ & s' & \vdots \\ \mathbf{0}^H & \widehat{\sigma}' & s \\ \mathbf{0}^H & 0 & \widehat{\sigma} \end{bmatrix}$, where $s$ and $s'$ are small numbers; | |
| update the subspace $[\mathbf{V}_s(t_k)\ \mathbf{v}_n]$ to become $[\mathbf{V}_s'\ \mathbf{V}_r'\ \mathbf{v}_n']$. | |
| $\widehat{\sigma}_n^2(t_k) = \frac{(M-P-1)(\beta\widehat{\sigma}_n(t_{k-1}))^2 + \|\widehat{\sigma}\|^2 + \|\widehat{\sigma}'\|^2}{M-P+1}$ | |
| $\mathbf{V}_s(t_k) = \mathbf{V}_s'$; $P = P - 1$. | |
| else $\widehat{\sigma}_n^2(t_k) = \frac{(M-P-1)(\beta\widehat{\sigma}_n(t_{k-1}))^2 + \widehat{\sigma}^2}{M-P}$ | |
| % Go to $t_k = t_{k+1}$ | |

during interval $[t_k, t_{k+1})$, and, after time $t_{k+1}$, no rank change occurs. Three situations regarding rank change can occur:

1. Rank increases by one (a new signal emerges during $[t_k, t_{k+1})$): The full-rank $M \times P$ steering matrix $\mathbf{A}(t_k)$ becomes the $M \times (P+1)$ steering matrix $\mathbf{A}(t_n) \triangleq [\mathbf{a}_1(t_n), \cdots, \mathbf{a}_P(t_n), \mathbf{a}_{P+1}(t_n)]$ $(\forall n > k)$ in which $\mathbf{a}_{P+1}(t_n)$ is the array response of the newly-appeared signal. Component $s_{P+1}(t_n)\mathbf{a}_{P+1}(t_n)$ $(\forall n > k)$ in (4.4) provides rank-increase information, and the remaining components provide refreshing information of the old $P$ signals for the tracking algorithm. If signal $s_{P+1}(t)$ remains present, then, at some later time instant $t_d > t_{k+1}$, the tracking algorithm detects its presence and increases the rank estimate of $\mathbf{A}(t_d)$ by one. Time lag, $t_d - t_{k+1}$, is a measure used to evaluate the detection ability for a tracking algorithm.

2. Rank decreases by one (one of the signals disappears during $[t_k, t_{k+1})$): The $M \times P$ steering matrix $\mathbf{A}(t_k)$ becomes the $M \times (P-1)$ steering matrix $\mathbf{A}(t_n) \triangleq [\mathbf{a}_1(t_n), \cdots, \mathbf{a}_{P-1}(t_n)]$ $(\forall n > k)$ in which the array response of the disappeared signal, $\mathbf{a}_P(t_n)$, is removed. The tracking algorithm uses rank-degenerate measurements $\mathbf{r}(t_n)$ $(\forall n > k)$ until at some instant $t_d > t_{k+1}$, the tracking algorithm detects rank degeneracy and decreases the rank estimate by one. Time lag, $t_d - t_{k+1}$, is also a measure used to evaluate the detection ability for an algorithm.

3. Rank remains unchanged (no signal emerges or disappears): Each measurement $\mathbf{r}(t_n)$, $(\forall n > k)$, has components in all dimensions $\{\mathbf{a}_1(t_n), \cdots, \mathbf{a}_M(t_n)\}$ because of nonzero $\mathbf{s}(t_n)$. Each new measurement $\mathbf{r}(t_n)$, $(\forall n > k)$, keeps providing *refreshing* information, and the tracking algorithm continuously updates the desired eigenstructure.

Next, let us see how each newly-arrived virtual snapshot $\mathbf{c}_{muv}(t_n)$, where $m, u, v$ are fixed and $\forall n > k$, provides similar update information about $\mathbf{A}(t_n)$ $(\forall n > k)$ for a tracking algorithm. Consider the same three situations after time $t_k$:

1. Rank increases by one during $[t_k, t_{k+1})$: According to (5.3) and (5.4), the full-rank $M \times P$ steering matrix $\mathbf{A}(t_k)$ becomes the $M \times (P+1)$ steering matrix

$\mathbf{A}(t_n) \triangleq [\mathbf{a}_1(t_n), \cdots, \mathbf{a}_P(t_n), \mathbf{a}_{P+1}(t_n)]$, $(\forall n > k)$, in which $\mathbf{a}_{P+1}(t_n)$ is the array response of the newly-appeared signal. Virtual snapshot $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ contains rank increase information
$\gamma_{4,s_{P+1}}(t_n)A_{m,P+1}(t_n)A^*_{u,P+1}(t_n)A^*_{v,P+1}(t_n)\mathbf{a}_{P+1}(t_n)$. We can expect
$\gamma_{4,s_{P+1}}(t_n)A_{m,P+1}(t_n)A^*_{u,P+1}(t_n)A^*_{v,P+1}(t_n)\mathbf{a}_{P+1}(t_n)$ to play the same role as $s_{P+1}(t_n)\mathbf{a}_{P+1}(t_n)$ in data-matrix-based tracking methods.

2. Rank decreases by one during $[t_k, t_{k+1})$: It is very easy to see that all virtual measurements $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ carry the same rank-degeneracy information as $\mathbf{r}(t_n)$.

3. Rank remains unchanged: This is the case where random $\mathbf{r}(t)$ is different from deterministic $\mathbf{c}_{muv}(t)$, i.e., if we use $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ we may lose tracking information. It is easier to see the problem for a stationary environment, i.e., when $\mathbf{A}(t)$ and $\gamma_{4,s_p}(t)$, $\forall p$ are constants. In a stationary environment, from (5.3) and (5.4), $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ is constant, since $\mathbf{A}(t_n)$ and $\tilde{\mathbf{s}}_{muv}(t_n)$ $(\forall n > k)$ are constants; therefore, $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ provides rank-1 information. When replacing $\mathbf{r}(t_n)$ by $\mathbf{c}_{muv}(t_n)$ $(\forall n > k)$ in any data-matrix-based subspace tracker, the tracked P-rank subspace gradually reduces to a rank-1 subspace spanned by constant $\mathbf{c}_{muv}(t_n)$, and full P-ranked subspace information is lost. This was not a problem for the actual snapshots, since $[\mathbf{r}(t_k), \mathbf{r}(t_{k+1}), \cdots] = \mathbf{A}(t_k)[\mathbf{s}(t_k), \mathbf{s}(t_{k+1}), \cdots]$ provides full-rank information, as long as $\mathbf{A}(t_n)$ $(\forall n > k)$ is of full rank and constant, and $[\mathbf{s}(t_k), \mathbf{s}(t_{k+1}), \cdots]$ is of full rank. It is easy for $[\mathbf{s}(t_k), \mathbf{s}(t_{k+1}), \cdots]$ to be of full rank because $\mathbf{s}(t)$ is a random vector. Consequently, we conclude that actual snapshots, $\mathbf{r}(t)$, provide full-rank subspace information, whereas virtual snapshots, $\mathbf{c}_{muv}(t)$, provide rank-degeneracy information.

From the above comparisons, we observe that in the situation when the rank of the signal subspace changes, actual snapshots and virtual snapshots both provide update information; however, in the situation when rank remains unchanged, actual snapshots provide full-rank information, but virtual snapshots do not. A cure to the problem of using rank-1 virtual snapshots is to use $P$ virtual snapshots, e.g.,

$\{\mathbf{c}_{mu1}(t_n), \cdots, \mathbf{c}_{muP}(t_n)\}$ with fixed $m$ and $u$, for each iteration, since they are linearly independent and span $\mathcal{R}(\mathbf{A}(t))$. If we substitute these P virtual snapshots for actual snapshots, one virtual snapshot at a time, into already-developed algorithms, then we have to repeat such replacements P times to complete one iteration. Consequently, we may end up with P times the complexity of the data-matrix-based algorithm, in addition to the computations of P virtual snapshots. For example, if the data-matrix-based algorithm is of complexity $\mathcal{O}(MP)$, then the corresponding cumulant-based algorithm, using P virtual snapshots for each iteration, is of complexity $\mathcal{O}(MP^2)$. The computational complexity $\mathcal{O}(MP^2)$ is still too costly.

To obtain a subspace tracker of complexity $\mathcal{O}(MP)$, we propose the following cyclic implementation of the P virtual snapshots. Suppose $\{\mathbf{c}_{mu1}(t), \cdots, \mathbf{c}_{muP}(t)\}$, where $m$ and $u$ are fixed, are the P independent virtual snapshots to be substituted for actual snapshot $\mathbf{r}(t)$ into, e.g., Rabideau's rank-adaptive fast subspace tracker (RA-FST) (see Table 5.1), and they are updated at each iteration. At time $t_k$, compute $\{\mathbf{c}_{mu1}(t_k), \cdots, \mathbf{c}_{muP}(t_k)\}$ and apply $\mathbf{c}_{mu1}(t_k)$, instead of $\mathbf{r}(t_k)$, for RA-FST; at time $t_{k+1}$, compute $\{\mathbf{c}_{mu1}(t_{k+1}), \cdots, \mathbf{c}_{muP}(t_{k+1})\}$ and apply $\mathbf{c}_{mu2}(t_{k+1})$, instead of $\mathbf{r}(t_{k+1})$; $\cdots$; and, at time $t_{k+P-1}$, compute $\{\mathbf{c}_{mu1}(t_{k+P-1}), \cdots, \mathbf{c}_{muP}(t_{k+P-1})\}$ and apply $\mathbf{c}_{muP}(t_{k+P-1})$. Doing this completes one cycle of the algorithm that uses P virtual snapshots; so, one cycle includes P iterations. A new cycle starts at time $t_{k+P}$; we compute $\{\mathbf{c}_{mu1}(t_{k+P}), \cdots, \mathbf{c}_{muP}(t_{k+P})\}$ and apply $\mathbf{c}_{mu1}(t_{k+P})$; at time $t_{k+P+1}$, we compute $\{\mathbf{c}_{mu1}(t_{k+P+1}), \cdots, \mathbf{c}_{muP}(t_{k+P+1})\}$ and apply $\mathbf{c}_{mu2}(t_{k+P+1})$; $\cdots$; and so on.

The computation of all $\{\mathbf{c}_{mu1}(t), \cdots, \mathbf{c}_{muP}(t)\}$ for each iteration is not a requirement, but it simplifies the update procedure: the update of $\{\mathbf{c}_{mu1}(t), \cdots, \mathbf{c}_{muP}(t)\}$ only involves the newly arrived snapshot, and, if rectangular windowing is used, an old snapshot.

Because the RA-FST only processes a single virtual snapshot for each iteration, the resulting cumulant-based subspace tracker can avoid the rank-degeneracy problem and achieves low computational complexity in tracking the time-varying signal subspace $\mathcal{R}(\mathbf{A}(t))$.

In summary, our new method preprocesses the actual measurements, i.e., for each iteration, it uses an actual measurement to generate a virtual measurement, which can then be used by any data-matrix-based subspace tracker. Our preprocessing

is comprised of two parts: computation of the P virtual measurements and cyclic implementation.

## 5.3 Implementation Considerations

In practice, we can not use true statistics $\mathbf{c}_{muv}(t)$; instead, we estimate $\mathbf{c}_{muv}(t)$ from a block of data snapshots $\{\mathbf{r}(t_{k-N+1}), \cdots, \mathbf{r}(t_k)\}$, i.e.,

$$\begin{aligned}
\widehat{\mathbf{c}}_{muv}(t_k) \triangleq & \frac{1}{N}\sum_{i=k-N+1}^{k} r_m(t_i)r_u^*(t_i)r_v^*(t_i)\mathbf{r}(t_i) \\
& - (\frac{1}{N}\sum_{i=k-N+1}^{k} r_m(t_i)r_u^*(t_i))\,(\frac{1}{N}\sum_{i=k-N+1}^{k} r_v^*(t_i)\mathbf{r}(t_i)) \\
& - (\frac{1}{N}\sum_{i=k-N+1}^{k} r_m(t_i)r_v^*(t_i))\,(\frac{1}{N}\sum_{i=k-N+1}^{k} r_u^*(t_i)\mathbf{r}(t_i)) \\
& - (\frac{1}{N}\sum_{i=k-N+1}^{k} r_u^*(t_i)r_v^*(t_i))\,(\frac{1}{N}\sum_{i=k-N+1}^{k} r_m(t_i)\mathbf{r}(t_i))
\end{aligned} \tag{5.5}$$

This estimator of $\widehat{\mathbf{c}}_{muv}(t_k)$ converges in probability to $\mathbf{c}_{muv}(t_k)$ for stationary and zero-mean $\mathbf{r}(t)$ [20, 52], i.e.,

$$\widehat{\mathbf{c}}_{muv}(t_k) \xrightarrow{p} \mathbf{c}_{muv}(t_k). \tag{5.6}$$

In a time-varying environment, there are two windowing strategies for updating quantities of interest: rectangular windowing and exponential windowing [100]. Sliding rectangular windowing suggests the so-called rank-2 update formula for a correlation matrix,

$$\widehat{\mathbf{R}}(t_{k+1}) = \widehat{\mathbf{R}}(t_k) + \frac{1}{N}\mathbf{r}(t_{k+1})\mathbf{r}^H(t_{k+1}) - \frac{1}{N}\mathbf{r}(t_{k-N})\mathbf{r}^H(t_{k-N}) \tag{5.7}$$

Table 5.2: Computations of virtual snapshot $\widehat{\mathbf{c}}_{muv}(t)$ from actual measurements using a sliding rectangular window.

| Update | Flops |
|---|---|
| $\widehat{Mom}(t_{k+1})_{M\times 1} = \widehat{Mom}(t_k) + \frac{1}{N} r_m(t_{k+1}) r_u^*(t_{k+1}) r_v^*(t_{k+1}) \mathbf{r}(t_{k+1})$ $-\frac{1}{N} r_m(t_{k-N+1}) r_u^*(t_{k-N+1}) r_v^*(t_{k-N+1}) \mathbf{r}(t_{k-N+1})$ | $2M+6$ |
| $\widehat{cov}_{11}(t_{k+1})_{1\times 1} = \widehat{cov}_{11}(t_k) + \frac{1}{N} r_m(t_{k+1}) r_u^*(t_{k+1})$ $-\frac{1}{N} r_m(t_{k-N+1}) r_u^*(t_{k-N+1})$ | 4 |
| $\widehat{cov}_{12}(t_{k+1})_{M\times 1} = \widehat{cov}_{12}(t_k) + \frac{1}{N} r_v^*(t_{k+1}) \mathbf{r}(t_{k+1})$ $-\frac{1}{N} r_v^*(t_{k-N+1}) \mathbf{r}(t_{k-N+1})$ | $2M+2$ |
| $\widehat{cov}_{13}(t_{k+1})_{M\times 1} = \widehat{cov}_{13}(t_k) + \frac{1}{N} r_u^*(t_{k+1}) \mathbf{r}(t_{k+1})$ $-\frac{1}{N} r_u^*(t_{k-N+1}) \mathbf{r}(t_{k-N+1})$ | $2M+2$ |
| $\widehat{cov}_{14}(t_{k+1})_{1\times 1} = \widehat{cov}_{14}(t_k) + \frac{1}{N} r_m(t_{k+1}) r_v^*(t_{k+1})$ $-\frac{1}{N} r_m(t_{k-N+1}) r_v^*(t_{k-N+1})$ | 4 |
| $\widehat{cov}_{15}(t_{k+1})_{M\times 1} = \widehat{cov}_{15}(t_k) + \frac{1}{N} r_m(t_{k+1}) \mathbf{r}(t_{k+1})$ $-\frac{1}{N} r_m(t_{k-N+1}) \mathbf{r}(t_{k-N+1})$ | $2M+2$ |
| $\widehat{cov}_{16}(t_{k+1})_{1\times 1} = \widehat{cov}_{16}(t_k) + \frac{1}{N} r_u^*(t_{k+1}) r_v^*(t_{k+1})$ $-\frac{1}{N} r_u^*(t_{k-N+1}) x_v^*(t_{k-N+1})$ | 4 |
| $\widehat{\mathbf{c}}_{muv}(t_{k+1}) = \widehat{Mom}(t_{k+1}) - \widehat{cov}_{11}(t_{k+1}) \widehat{cov}_{12}(t_{k+1})$ $-\widehat{cov}_{13}(t_{k+1}) \widehat{cov}_{14}(t_{k+1}) - \widehat{cov}_{15}(t_{k+1}) \widehat{cov}_{16}(t_{k+1})$ | $3M$ |
| % Update $\widehat{\mathbf{c}}_{muv}(t_{k+1})$ | $11M+22$ |

where an update is associated with $\mathbf{r}(t_{k+1})$ and a downdate is associated with $\mathbf{r}(t_{k-N})$. Exponential windowing suggests the rank-1 update formula for a correlation matrix,

$$\widehat{\mathbf{R}}(t_{k+1}) \;=\; \beta\, \widehat{\mathbf{R}}(t_k) + (1-\beta)\, \mathbf{r}(t_{k+1}) \mathbf{r}^H(t_{k+1}) \tag{5.8}$$

where $0 < \beta < 1$ is the forgetting factor.

The advantage for exponential windowing is that it requires only a rank-1 update; its disadvantage is that forgetting factor $\beta$ must be determined appropriately; otherwise, the influence of old data may last for a long time when $\beta$ is large, or the influence of the new data may fade away quickly when $\beta$ is small. The advantage of a short rectangular window is that it can detect a sudden change in a signal, whereas its disadvantage is that it requires more computations (rank-2 update).

Table 5.3: Computations of virtual snapshot $\widehat{\mathbf{c}}_{muv}(t)$ from actual measurements using an exponential window.

| Update | Flops |
|---|---|
| $\widehat{Mom}(t_{k+1})_{M\times 1} = \beta_1 \widehat{Mom}(t_k)$ $+(1-\beta_1) r_m(t_{k+1}) r_u^*(t_{k+1}) r_v^*(t_{k+1}) \mathbf{r}(t_{k+1})$ | $2M+3$ |
| $\widehat{cov}_{11}(t_{k+1})_{1\times 1} = \beta_2 \widehat{cov}_{11}(t_k) + (1-\beta_2) r_m(t_{k+1}) r_u^*(t_{k+1})$ | 3 |
| $\widehat{cov}_{12}(t_{k+1})_{M\times 1} = \beta_2 \widehat{cov}_{12}(t_k) + (1-\beta_2) r_v^*(t_{k+1}) \mathbf{r}(t_{k+1})$ | $2M+1$ |
| $\widehat{cov}_{13}(t_{k+1})_{M\times 1} = \beta_2 \widehat{cov}_{13}(t_k) + (1-\beta_2) r_u^*(t_{k+1}) \mathbf{r}(t_{k+1})$ | $2M+1$ |
| $\widehat{cov}_{14}(t_{k+1})_{1\times 1} = \beta_2 \widehat{cov}_{14}(t_k) + (1-\beta_2) r_m(t_{k+1}) r_v^*(t_{k+1})$ | 3 |
| $\widehat{cov}_{15}(t_{k+1})_{M\times 1} = \beta_2 \widehat{cov}_{15}(t_k) + (1-\beta_2) r_m(t_{k+1}) \mathbf{r}(t_{k+1})$ | $2M+1$ |
| $\widehat{cov}_{16}(t_{k+1})_{1\times 1} = \beta_2 \widehat{cov}_{16}(t_k) + (1-\beta_2) r_u^*(t_{k+1}) r_v^*(t_{k+1})$ | 3 |
| $\widehat{\mathbf{c}}_{muv}(t_{k+1}) = \widehat{Mom}_1(t_{k+1}) - \widehat{cov}_{11}(t_{k+1}) \widehat{cov}_{12}(t_{k+1})$ $-\widehat{cov}_{13}(t_{k+1}) \widehat{cov}_{14}(t_{k+1}) - \widehat{cov}_{15}(t_{k+1}) \widehat{cov}_{16}(t_{k+1})$ | $3M$ |
| % Update $\widehat{\mathbf{c}}_{muv}(t_{k+1})$ | $\underline{11M+15}$ |

These two windowing schemes can also be applied for the estimator $\widehat{\mathbf{c}}_{muv}(t_{k+1})$ [12, 33]. In Table 5.2, the update of $\widehat{\mathbf{c}}_{muv}(t_{k+1})$ using the rectangular window includes a rank-2 update for the fourth-order moment, a rank-2 update for each of the six cross-correlation terms, and three multiplications of cross-correlation terms. In Table 5.3, the update of $\widehat{\mathbf{c}}_{muv}(t_{k+1})$ using the exponential window includes a rank-1 update for the fourth-order moment, a rank-1 update for each of the six cross-correlation term, and three multiplications of cross-correlation terms. Updating moment terms requires much more computations than those for updating correlation terms. Note that, in Table 5.3, the forgetting factor $\beta_1$ for the fourth-order moment estimator is distinct from the forgetting factor $\beta_2$ for the cross-correlation estimators, i.e., $\beta_1 \neq \beta_2$. This is due to the fact that the estimator for a fourth-order moment has a longer effective length than the estimator for a cross correlation. In our simulations, we found that rectangular windowing yields more stable results; hence, we recommend using the rectangular window.

## 5.4 Adaptive Virtual-ESPRIT for Target Tracking

Our proposed preprocessing method can be combined with any data-matrix-based fast subspace tracker. Among various $\mathcal{O}(MP)$ algorithms, RA-FST preserves signal eigenstructure, which allows us to do both rank tracking and subspace tracking.

For illustration purposes, our simulations are for the target and rank tracking of independent signals using VESPA(5), which is VESPA applied to the 5-element array configuration in Fig. 3.2(a). Recall that a full batch-processed TLS-VESPA requires two SVDs and one EVD [50]. Focusing on subspace tracking, our *adaptive VESPA(5)* only replaces the first SVD, which tracks the signal rank and subspace, by the cumulant-based RA-FST; the second SVD and last EVD are performed non-adaptively. Note that the cumulant-based preprocessing can also be used with extended-VESPA [29] to track the *generalized* signal subspace in the presence of coherent signals.

Recall that VESPA(5) creates a virtual copy of the main array (see Fig. 3.2(a) and [13]). Let $\widehat{\mathbf{c}}_{m1v}(t)$ and $\widehat{\mathbf{c}}_{m2v}(t)$, $\forall m, v \in \{1, 2, \cdots, 5\}$, be the virtual snapshots associated with the main array and the virtual copy of the main array, respectively. In the presence of P signals, our cumulant-based preprocessing updates a total of 2P virtual snapshots for each iteration, i.e., $\{\widehat{\mathbf{c}}_{m11}(t), \cdots, \widehat{\mathbf{c}}_{m1P}(t)\}$ and $\{\widehat{\mathbf{c}}_{m21}(t), \cdots, \widehat{\mathbf{c}}_{m2P}(t)\}$ for an arbitrarily chosen $m$, $m \in \{1, \cdots, 5\}$; but, only one pair $[\widehat{\mathbf{c}}^T_{m1v}(t)\ \widehat{\mathbf{c}}^T_{m2v}(t)]^T$, $v \in \{1, \cdots, P\}$, is processed by RA-FST as explained in Section 5.2. We call the complete algorithm "adaptive-VESPA(5)"; it includes cumulant-based preprocessing, RA-FST, a second non-adaptive SVD, and finally an EVD. Adaptive-VESPA(5) tracks the subspace of the virtually-extended array configuration as depicted in Fig. 3.2(a).

It is straightforward to produce an adaptive-ESPRIT(8) for the array ESPRIT(8) in Fig. 3.2(b): a pair of actual snapshots from the two identical subarrays is used directly instead of the cumulant-based virtual snapshots; everything else is the same as in adaptive-VESPA(5).

Obviously, adaptive-VESPA(5) has a hardware saving advantage over adaptive-ESPRIT(8); namely, adaptive-VESPA(5) uses a 5-element array to track up to 4 targets, whereas adaptive-ESPRIT(8) uses two 4-element arrays to track up to 3 targets.

## 5.5 Discussions

In Section 5.4, the use of 2P virtual snapshots by adaptive-VESPA(5) is a special case of applying cumulants to ESPRIT. Here we discuss a wider range of applying cumulants to subspace methods, e.g., cumulant-based MUSIC is also covered [29]; hence, we focus on tracking $\mathcal{R}(\mathbf{A}(t))$. Our method can use $P$ virtual snapshots, $\{\mathbf{c}_{mu1}(t)\ \mathbf{c}_{mu2}(t)\cdots\mathbf{c}_{muP}(t)\}$, along with MUSIC to track targets; also, our method can use $2P$ virtual snapshots along with ESPRIT, which was explained in Section 5.4.

From Section 5.2, observe that we have three degrees of freedom in choosing the subscripts of $\mathbf{c}_{muv}(t)$, i.e., $m,u,v \in \{1,\cdots,M\}$; but, we only used the *minimum redundant* P virtual snapshots to achieve our goal. Minimum redundancy refers to the smallest number of virtual snapshots that retain signal subspace information. We can use more virtual snapshots to achieve better performance, something suggested in [29] to improve performance by choosing multiple guiding sensor pairs to compute more cumulant matrices. Of course, the cost to do this is more computations.

One way to visualize the virtual snapshots is that they are columns taken from the $M \times M$ cumulant matrix $[\mathbf{c}_{mu1}(t)\,\mathbf{c}_{mu2}(t)\cdots\mathbf{c}_{muM}(t)]$ (fixed $m$ and $u$), as described in [29]. In batch processing, we have no idea in advance about the number of signals, and the entire cumulant matrix has to be computed. In a real-time implementation, the number of signals is known, and is calculated from the preceding iteration; hence, we only need to update the minimum redundant submatrix of the entire cumulant matrix.

A sufficient condition for the cumulant-based tracking algorithm to work is that the P virtual snapshots must be linearly independent. Does this condition impose new constraints, in addition to that of a pair of identical sensors, on the cumulant-based tracking algorithm ? Applying (5.3) and (5.4), using the stationarity assumption that $\mathbf{A}(t)$ and $\gamma_{4,s_p}(t)$ are constants, we have

$$\begin{aligned}
&[\mathbf{c}_{mu1}(t_k)\ \mathbf{c}_{mu2}(t_{k+1})\cdots\mathbf{c}_{muP}(t_{k+P-1})] \\
= \ &\mathbf{A}(t_k)\ \mathrm{diag}\{A_{m1}(t_k)A^*_{u1}(t_k)\gamma_{4,s_1}(t_k),\cdots,A_{mP}(t_k)A^*_{uP}(t_k)\gamma_{4,s_P}(t_k)\} \\
&\begin{bmatrix} A_{11}(t_k) & \cdots & A_{1P}(t_k) \\ \vdots & \ldots & \vdots \\ A_{P1}(t_k) & \cdots & A_{PP}(t_k) \end{bmatrix}^H
\end{aligned} \tag{5.9}$$

From (4.2), $A(m,u;t_k) \neq 0$, $\forall m,u$; from (4.3), $\gamma_{4,s_p}(t_k) \neq 0$, $\forall p$; hence, the diagonal matrix in (5.9) has nonzero diagonal entries. The matrix inside the brackets of (5.9) is the $P \times P$ principal submatrix of full-rank $\mathbf{A}(t_k)$, and is also of full rank. Combining these facts, we conclude that using these P virtual snapshots does not add new constraints to the tracking algorithm.

## 5.6 Experiments

In this section, we describe simulation results for target and rank tracking using the adaptive-VESPA(5) introduced in Section 5.4. In both Sections 5.6.1 and 5.6.2, two signals are present and are of the same power. We use a single SNR value to indicate the SNR condition between the multiple signals and noises.

In computing virtual snapshots, we used a sliding rectangular window of size $N$. The RA-FST used an exponential window, with a forgetting factor of $\beta$. Thresholds for hypothesis tests in RA-FST to determine rank increase and rank decrease are denoted by $T_{\text{inc}}$ and $T_{\text{dec}}$, respectively (see Table 5.1).

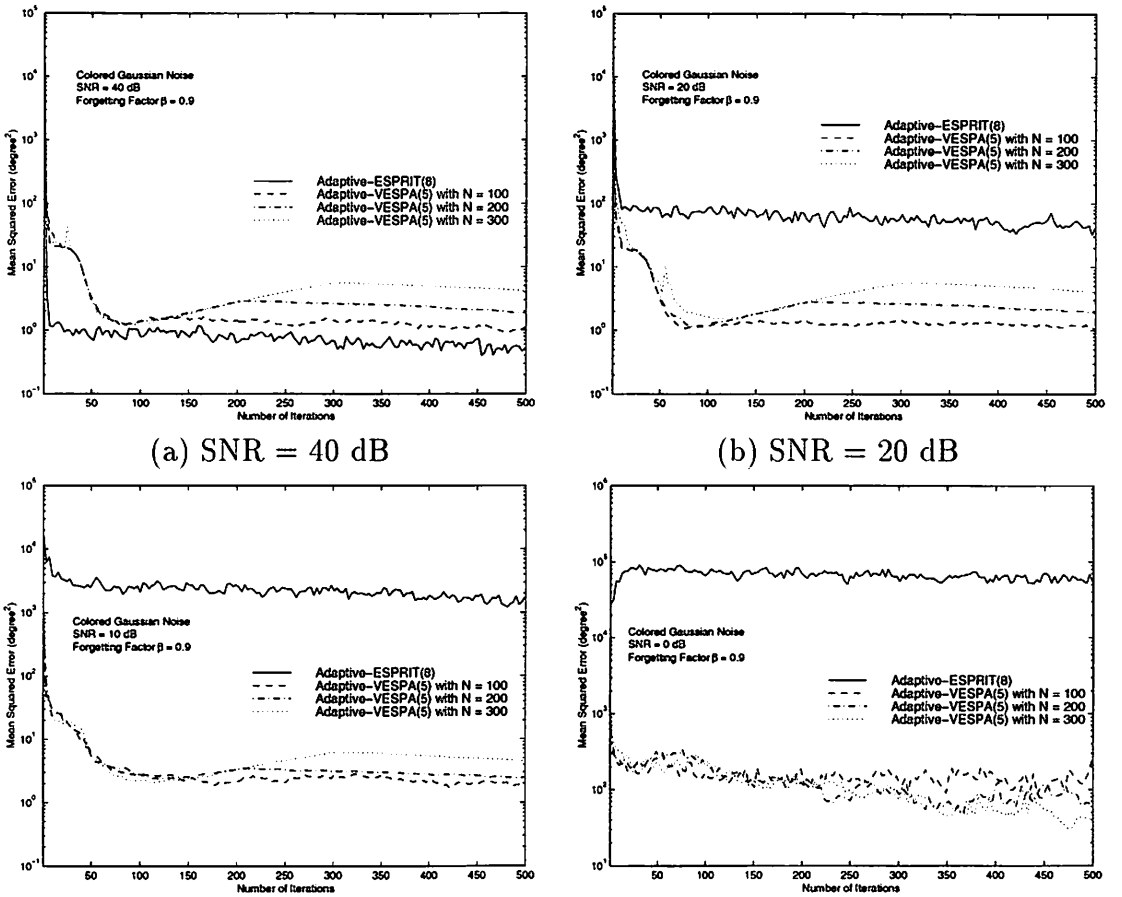
### 5.6.1 Target Tracking

Our first experiment compares the target tracking ability of adaptive-VESPA(5) and adaptive-ESPRIT(8) (described in Section 5.4) under additive colored Gaussian noise. Because VESPA(5) is a subarray of ESPRIT(8), adaptive-VESPA(5) uses partial measurements from those used by adaptive-ESPRIT(8). Within the observation interval, two signals are moving along two sinusoidal trajectories,

$$\begin{aligned} \phi_1(t_k) &= 0^\circ + 10^\circ \sin(2\pi t_k/4000) \\ \phi_2(t_k) &= 10^\circ + 10^\circ \sin(2\pi t_k/4000) \end{aligned} \qquad t_k = 1, 2, \cdots \tag{5.10}$$

Adaptive-ESPRIT(8) and adaptive-VESPA(5) [with three different window sizes ($N = 100, 200$, and $300$)] were applied for each of the SNR conditions, SNR = 40, 20, 10, 0 dB. We assume that the number of signals is known and is fixed all the time in both adaptive-VESPA(5) and adaptive-ESPRIT(8); therefore, the threshold tests to detect rank changes are omitted from our algorithms. The tracking MSEs

[see (4.35)], averaged over 100 MC trials at each $t_k$, versus the number of iterations are plotted in Fig. 5.1.




(a) SNR = 40 dB (b) SNR = 20 dB

(c) SNR = 10 dB (d) SNR = 0 dB

Figure 5.1: Mean squared error of target tracking by adaptive-ESPRIT(8) and adaptive-VESPA(5), with three different window sizes, versus the number of iterations.

For the high SNR (40 dB) case of Fig. 5.1(a), colored noise has little influence on the adaptive-ESPRIT(8) results, and adaptive-ESPRIT(8) outperforms the three adaptive-VESPA(5)s in terms of faster convergence and lower steady-state MSEs. Because adaptive-VESPA(5) uses rectangular windowing in estimating virtual snapshots, it converges slowly and, therefore, has a larger MSE. Obviously, the MSEs are mainly caused by the time-varying environment in this case.

As the SNR decreases, adaptive-ESPRIT(8) still converges quickly; but, it converges to wrong values and suffers from high MSEs as shown in Figs. 5.1(b)-(d). On

the other hand, results from the three adaptive-VESPA(5)s have fairly low steady-state MSEs, because cumulants are robust to colored Gaussian noise.

The window size also has influence on the adaptive-VESPA(5). In Figs. 5.1(a)-(c), adaptive-VESPA(5) with a shorter (N=100) window size has smaller steady-state MSEs than those with longer (N=200 and 300) window sizes. This is due to the fact that adaptive-VESPA(5) with a longer window size is less sensitive to signal changes. Furthermore, Fig. 5.1(d) shows that adaptive-VESPA(5) with a longer window size (N=300) is more robust to low SNRs than adaptive-VESPA(5) with the shorter window sizes.

From this experiment, we conclude that adaptive-VESPA(5) is suitable for tracking slowly time-varying signals for medium SNR conditions.

### 5.6.2 Rank Tracking

In our second experiment, we demonstrate rank tracking by adaptive-VESPA(5). We assume that during the observation interval between time 0 and 1500, the first signal, at direction 0°, was present all the time, and that the second signal, at direction 10°, was present from time 0 to 500, disappeared at 501, and reappeared at time 1001; therefore, there is a rank decrease at time 500, and a rank increase at 1001. Parameters for adaptive-VESPA(5) are: SNR = 10 dB, $N = 300$, $\beta = 0.95$, $T_{\text{inc}} = 0.006$, and $T_{\text{dec}} = 0.015$ (see Table 5.1).

For each MC trial, adaptive-VESPA(5) is said to have *successful* rank tracking if a single decrease in rank and a single increase in rank are detected sometime after times 500 and 1000, respectively. The *success rate* of rank tracking is the number of MC trials that have successful rank tracking divided by the total number of MC trials.

Adaptive-VESPA(5) has a success rate of 76%, i.e., 76 successes out of 100 MC trials. Sample means of the time lags for successful tracking of rank decrease and increase are 138.51 and 208.29, respectively; sample STDs of the time lags for successful tracking of rank decrease and increase are 30.57 and 36.90, respectively. These statistical numbers, again, suggest that adaptive-VESPA(5) is suitable for slowly time-varying environments.

Five MC realizations of *successful* rank tracking are plotted in Fig. 5.2. Degraded tracking occurs whenever there is rank mismatch between true rank and estimated rank, as shown by the tracking trajectories during time periods after time 500 and 1000. Sometimes, this rank mismatch may cause very poor tracking results, as shown by the top trajectory around time 1200.

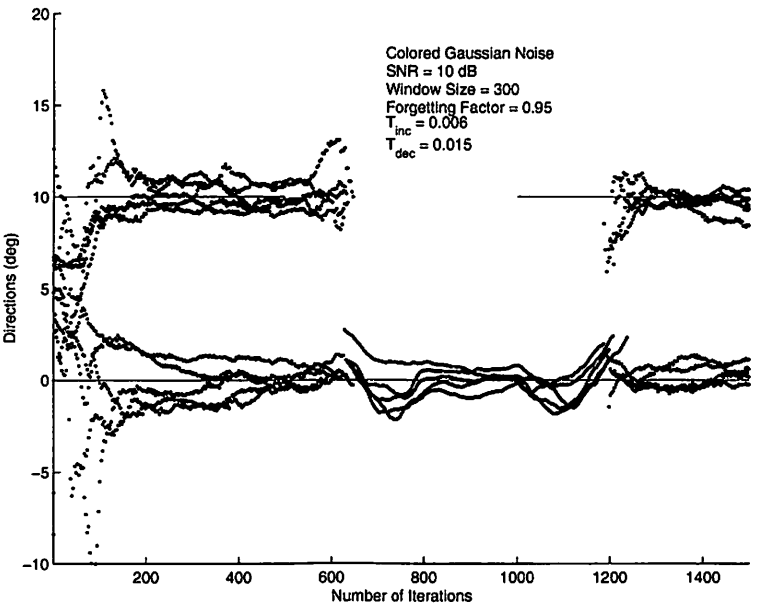



Figure 5.2: Five realizations of rank tracking by adaptive-VESPA(5).

Theoretically, the singular values, $\sigma_n(t)$, for singular vectors in the noise subspace are zero. Practically, an estimate, $\widehat{\sigma}_n(t)$, depends on a lot of factors, including array manifold, rectangular window size, forgetting factor, and SNR. If $T_{\text{inc}}$ is too small, we run the risk of a high probability of false alarm, i.e., there is no actual rank increase, but we think there is a rank increase. If $T_{\text{inc}}$ is too large, we run the risk of a high probability of miss, i.e., there is a rank increase, but we think no rank increase occurs. Similar difficulties occur in the rank-decrease test. Although the threshold tests to determine rank increase and decrease are simple, their analyses to compute useful information regarding probability of detection, probability of miss, and probability of false alarm is more difficult for cumulant-based methods than it is for correlation-based methods, and involves asymptotics, since previous performance analyses of cumulant methods are only based on asymptotics [5, 101]. We leave the analysis and determination of optimal $T_{\text{inc}}$ and $T_{\text{dec}}$ to future work.

Table 5.4: Results of rank tracking, based on 100 MC trials, by adaptive-VESPA(5) with different thresholds for each SNR condition.

| | SNR=40dB | SNR=20dB | SNR=10dB | SNR=5dB |
|---|---|---|---|---|
| $T_{\text{dec}}$ | 0.0145 | 0.015 | 0.015 | 0.032 |
| $T_{\text{inc}}$ | 0.0051 | 0.0051 | 0.006 | 0.02 |
| Success Rate | 94% | 90% | 76% | 32% |
| Sample Mean of the Time Lag to Detect Rank Decrease | 128.3 | 126.8 | 138.51 | 123.8 |
| Sample STD of the Time Lag to Detect Rank Decrease | 11.80 | 8.39 | 30.57 | 36.64 |
| Sample Mean of the Time Lag to Detect Rank Increase | 219.6 | 214.1 | 208.29 | 242.9 |
| Sample STD of the Time Lag to Detect Rank Increase | 32.06 | 30.68 | 36.90 | 94.84 |

Nevertheless, we repeated the above rank tracking experiments for SNR = 40, 20, and 5 dB; for each SNR, we applied adaptive-VESPA(5) with different thresholds, $T_{\text{dec}}$ and $T_{\text{inc}}$, as shown in Table 5.4. We performed many simulations and chose $T_{\text{dec}}$ and $T_{\text{inc}}$ that yielded the highest success rate for the corresponding SNR condition. Observe that the selected thresholds increase as the SNR decreases, because higher thresholds are selected to improve success rate when the SNR is low.

The first two columns of Table 5.4 contain about the same values, revealing that high-SNR colored noise has little influence on adaptive-VESPA(5). This fact is confirmed by the two curves, which correspond to the target tracking results obtained by adaptive-VESPA(5) for N = 300 and SNR = 40 and 20 dB, in Figs. 5.1(a) and (b). The success rate for SNR = 10 dB is lower than those for SNR = 40 and 20 dB, but, the sample means for the time lags to detect increase and decrease in rank do not change much. For SNR = 5 dB, success rate and sample STDs of the time lags to detect rank change are very poor. The poor tracking is due to the cumulant estimation that uses a short window size (N = 300).

## 5.7 Conclusions

We have developed a cumulant-based preprocessing method that can be combined with any data-matrix-based method to achieve low computational complexity of $\mathcal{O}(MP)$ for target and rank tracking. Its generalizations to other applications of array signal processing, such as cumulant-based adaptive beamforming, are straightforward. The resulting algorithm is suitable for a slowly time-varying environment. Although it has slower convergence than a data-matrix-based method (due to the nature of cumulants), it preserves all advantages of cumulant-based array signal processing methods, including hardware saving, greater robustness to colored noise, and greater robustness to model errors. Simulation results show this method works reasonably well for rank tracking and target tracking. In a colored noise environment, it significantly outperforms data-matrix-based target tracking for medium and low SNR situations.

# Chapter 6

# Subspace-Based Direction Finding Using Fractional Lower-Order Moments: MUSIC

A recent study [71] shows that stationary symmetric alpha stable ($S\alpha S$) processes are better models for impulsive noise than Gaussian processes. Real $S\alpha S$ processes belong to the larger class of stable distributions [69], denoted by $S_\alpha(\sigma, \beta, \mu)$, where $0 < \alpha \leq 2$ is the index of stability or characteristic exponent, $\sigma > 0$ is the scale parameter, $-1 \leq \beta \leq 1$ is the skewness parameter, and real-valued $\mu$ is the shift parameter. A univariate stable random variable $X$ is said to be $S\alpha S$ [71] if $\beta = 0$, i.e., $X \sim S_\alpha(\sigma, 0, \mu)$ [1], in which case $X$ is symmetrical about $\mu$. When $1 < \alpha \leq 2$, $\mu$ is the mean of the $S\alpha S$ random variable. Gaussian processes are stable processes with $\alpha = 2$.

A univariate $S\alpha S$ random variable $X$ has several interesting properties. A $S\alpha S$ random variable $X \sim S_\alpha(\sigma, 0, \mu)$ [$S_\alpha(\sigma, 0, \mu)$ is short for a closed-form probability density function (PDF)] is often characterized by its characteristic function, $E\{e^{jtX}\} = exp(-\sigma^\alpha |t|^\alpha + j\mu t)$. A $S\alpha S$ random variable X has finite *absolute p-th order moments* only for $0 < p < \alpha$, i.e., $E\{|X|^p\} < \infty$, $p < \alpha < 2$. This last property explains why second-order statistics are inappropriate for dealing with $S\alpha S$ processes.

A complex $S\alpha S$ random variable $X = X_1 + jX_2$ is characterized by its characteristic function $E\{e^{j(t_1 X_1 + t_2 X_2)}\} = exp(j(\mu_1 t_1 + \mu_2 t_2) - \sigma^\alpha |t_1^2 + t_2^2|^{\alpha/2})$. If $\mu_1 = \mu_2 = 0$,

[1] In [69], a univariate stable random variable is said to be $S\alpha S$ if $\beta = \mu = 0$, and the symmetry of $X$ is about the origin.

then the complex $X$ is said to be isotropic $S\alpha S$. Note that the marginal distributions of $X_1$ and $X_2$ are $S_\alpha(\sigma, 0, \mu_1)$ and $S_\alpha(\sigma, 0, \mu_2)$, respectively. A complex $S\alpha S$ random variable $X$ has finite fractional lower-order moment (FLOM) $p < \alpha < 2$, i.e., $E\{|X|^p\} < \infty$, $\forall p < \alpha < 2$. Obviously, $S\alpha S$ signals are of infinite power, because $E\{|X|^2\}$ is infinite. For details about $S\alpha S$ processes, see [69, 71] and the references therein.

The robust covariation-based MUSIC (ROC-MUSIC) [88] for direction finding used covariations and assumed that signals and additive noise are jointly $S\alpha S$. In fact, the covariation of two jointly $S\alpha S$ random variables can be expressed as a function of FLOMs when $1 < \alpha \leq 2$; therefore, ROC-MUSIC can also be viewed as a method based on FLOMs. We believe that the assumption made in [88] that both the signal and additive noise are jointly $S\alpha S$ is not realistic, because signals are always of finite power (power signals), whereas $S\alpha S$ signals are of infinite power. This motivates us to investigate direction finding in the presence of additive $S\alpha S$ noise using just FLOMs. In this chapter, we apply FLOMs to the scenarios where the signals are circular and the additive noise is $S\alpha S$ with $1 < \alpha \leq 2$. The resulting FLOM-based matrix can be used with MUSIC to retrieve the DOAs of signals.

The assumption that signals are of particular forms is commonly made in the literature on optimal DF. For example, signals are assumed to be Gaussian in maximum-likelihood based DF [73]. Subspace-based DF methods do not care about signal types; instead, they assume that the correlation matrix of the signals is non-singular. Computer simulations using circular signals can be found in [5, 77]. The assumption of circular signals is reasonable for active localization and some passive localization of underwater acoustics, and, some land mobile systems.

In Section 6.1, we define the problem of interest. In Section 6.2, we propose FLOM-based MUSIC to extract DOAs of the signals. Simulations are presented in Section 6.3. Finally, we give discussions and draw conclusions in Sections 6.4 and 6.5, respectively.

## 6.1 Problem Definition

Assume that there are $P$ narrowband independent signals impinging on an $M$-element array. Using the complex envelope representation, array output at the $k$-th sensor is

$$x_k(t) = \sum_{m=1}^{P} A_{km} s_m(t) + n_k(t), \qquad k = 1, 2, \cdots, M \tag{6.1}$$

where $A_{km}$ is the sensor response of the $k$-th sensor with respect to the $m$-th signal, $s_m(t)$ is the $m$-th signal measured at the reference point, and $n_k(t)$ is the additive noise at sensor $x_k$. Note that in (6.1), symbol that denotes the signal received at the $k$-th sensor has been changed from $r_k(t)$ to $x_k(t)$, because we think it will be clearer in notation when two subarrays are used [see Chapter 7]. The matrix form for (6.1) is

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t) + \mathbf{n}(t) \tag{6.2}$$

where:

| | |
|---|---|
| $\mathbf{x}_{M\times 1}(t) \triangleq [x_1(t), \cdots, x_M(t)]^T$ | measurement vector (snapshot); |
| $\mathbf{A}_{M\times P} \triangleq \{A_{km}\}$ | array manifold; |
| $\mathbf{s}_{P\times 1}(t) \triangleq [s_1(t), \cdots, s_P(t)]^T$ | signal vector; |
| $\mathbf{n}_{M\times 1}(t) \triangleq [n_1(t), \cdots, n_M(t)]^T$ | noise vector. |

The following assumptions are made throughout this chapter:

**[AS1]** $A_{km} \triangleq e^{-j\tau_{km}}$ and $\tau_{km} = 2\pi \frac{d_k}{\lambda} \sin(\phi_m)$, where $d_k$ is the distance between sensor $x_k$ and $x_1$ (the reference point), $\lambda$ is the wavelength of the carrier, and $\phi_m$ is the DOA of $s_m(t)$. This is a reasonable assumption for a calibrated array, where the sensor gains are known through calibration and are assumed to be unity.

**[AS2]** $s_m(t) = u_m(t)e^{j\psi_m(t)}$, where amplitude $u_m(t)$ and phase $\psi_m(t)$ are real processes that are statistically independent of each other, and $\psi_m(t)$ is a sequence of independent identically distributed (*i.i.d.*) random variables that are uniform over $[0, 2\pi)$. Signals of this form belong to the class of

circular signals; a complex random variable is said to be circular if its distribution is invariant under the multiplication by an arbitrary unit-modulus complex number [5]. If $u_m(t)$ is Rayleigh-distributed, then $s_m(t)$ is the circularly symmetrical Gaussian signal [59]. If $u_m(t)$ is a constant, $u_m$, then $s_m(t)$ is the phase modulated (PM) signal. These two examples show that the assumption of circular signals is practical for communication systems. This assumption also implies that $s_m(t)$ is zero-mean, because $e^{j\psi_m(t)}$ is zero-mean if $\psi_m(t)$ is uniform over $[0, 2\pi)$, and, $u_m(t)$ and $\psi_m(t)$ are statistically independent. Note that this assumption is crucial to our derivations.

**[AS3]** $n_k(t)$ is a sequence of *i.i.d.* isotropic complex $S\alpha S$ random variables with $1 < \alpha \leq 2$. This assumption implies that $n_k(t), \forall k$, is zero-mean because $n_k(t)$ is isotropic [69].

**[AS4]** Array manifold $\mathbf{A}$ is of full rank, which is required for all subspace methods to separate signal and noise subspaces.

Assumptions **[AS2]** and **[AS3]** together implies that $\mathbf{x}(t)$ is zero-mean. The problem of DOA estimation is to extract the DOAs, $\{\phi_p\}_{p=1}^{P}$, from measurements $\{\mathbf{x}(t)\}_{t=1}^{N}$.

## 6.2 Proposed Solution

In Theorem 6.1, we define a class of bounded matrices. Theses matrices are the basis for our FLOM-based MUSIC method. Let $\Re\{\cdot\}$ be the real part operator, $|\cdot|$ be the absolute value operator, and $E\{\cdot\}$ be the expectation operator.

**Theorem 6.1** *Let* $\mathbf{C}$ *be an* $M \times M$ *matrix, whose* $(i,k)$*-th entry* $C_{ik}$ *is defined as*

$$C_{ik} \triangleq E\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}, \qquad 1 < p < \alpha \leq 2 \tag{6.3}$$

*where* $x_i(t)$ *and* $x_k(t)$, $\forall i$ *and* $k$, *are the array outputs defined in Section 6.1. The* $C_{ik}$, $\forall i$ *and* $k$, *are bounded.*

**Proof**: See Appendix A.

Theorem 6.1 not only recommends FLOMs but also negates cross correlations, because the correlation of $x_i(t)$ and $x_k(t)$, which is a special case of $C_{ik}$ when $p = 2$, is unbounded. Boundedness is important for statistical signal processing, because statistical analyses cannot evaluate unbounded quantities to obtain reasonable conclusions. Note that the FLOM-based $\mathbf{C}$ is unbounded for $\alpha < 1$, and our method does not apply to such high impulsive environments. For $\alpha < 1$, the use of zero-memory non-linearity (ZMNL) [76] to clip the undesired impulsive noise may be a good solution.

Next, we show, in Theorem 6.2, that $\mathbf{C}$ is of a particular structure that is useful for DOA estimation. Note that a very important theorem, stated and proved in Appendix C, is used in the proof of Theorem 6.2.

**Theorem 6.2** *The $(i,k)$-th entry of $\mathbf{C}$, defined in Theorem 6.1, can be expressed as*

$$C_{ik} = \sum_{m=1}^{P} A_{im}\Lambda_{mm}A_{km}^{*} + \gamma\delta_{ik} \tag{6.4}$$

*where*

$$\Lambda_{mk} = \delta_{mk}\, E\{s_m(t)|\sum_{q=1}^{P} s_q(t) + n_m(t)|^{p-2}(\sum_{r=1}^{P} s_r(t) + n_m(t))^{*}\} \tag{6.5}$$

*and*

$$\gamma = E\{n_k(t)|\sum_{q=1}^{P} s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} s_r(t) + n_k(t))^{*}\}, \tag{6.6}$$

*in which $\delta_{mk}$ is the Kronecker delta. The matrix form for $\mathbf{C}$ is*

$$\mathbf{C} = \mathbf{A}\Lambda\mathbf{A}^{H} + \gamma\,\mathbf{I} \tag{6.7}$$

**Proof**: Substituting (6.1) into (6.3) for $x_i(t)$ and $x_k(t)$, we have

$$C_{ik} = \sum_{m=1}^{P} A_{im} \underbrace{E\{s_m(t)|\sum_{q=1}^{P} A_{kq}s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}s_r(t) + n_k(t))^*\}}_{F_{mk}}$$
$$+ \underbrace{E\{n_i(t)|\sum_{q=1}^{P} A_{kq}s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}s_r(t) + n_k(t))^*\}}_{D_{ik}} \quad (6.8)$$

The two underbraced terms in (6.8) are the same as the two defined in (C.2) and (C.1), respectively, with $A_{km} = e^{-j\tau_{km}}$ (see **[AS1]** and Appendix C). The substitution of (C.2) and (C.1) into (6.8) yields (6.4). □

Theorem 6.2 reveals that $\mathbf{C}$ includes the DOA information that is in the array manifold $\mathbf{A}$. Note that the correlation matrix of the array outputs, when additive white Gaussian noise is present, is $\mathbf{R}_{xx} = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_n^2\mathbf{I}$, where positive-definite $\mathbf{R}_s$ is the correlation matrix of signals and $\sigma_n^2$ is the variance of the white Gaussian noise. Comparing $\mathbf{R}_{xx} = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_n^2\mathbf{I}$ with $\mathbf{C}$ in (6.7), we can interpret $\Lambda$ as a fractional lower-order correlation matrix of the signals, and $\gamma$ as a fractional lower-order correlation of the additive noise level.

Theorem 6.3 reveals that the absolute FLOMs of the array outputs are identical, i.e., $C_{kk} \triangleq E\{|x_k(t)|^p\}$, $\forall k$, is constant. Although not useful at this moment, this fact will be used in Section 6.4.

**Theorem 6.3** *The diagonal entries of* $\mathbf{C}$ *are identical, i.e.,*

$$C_{11} = C_{22} = \cdots = C_{MM} \quad (6.9)$$

**Proof**: Using $A_{km} = e^{-j\tau_{km}}$ and $A_{km}A_{km}^* = 1$ (by **[AS1]**), we have from (6.4) that

$$C_{kk} = \sum_{m=1}^{P} \underbrace{A_{km}A_{km}^*}\Lambda_{mm} + \gamma$$
$$= \sum_{m=1}^{P} \Lambda_{mm} + \gamma \quad (6.10)$$

Note that the $C_{kk}$ in (6.10) is independent of index $k$, indicating (6.9) is true. □

To implement MUSIC on the $\mathbf{C}$ defined in (6.7), the $\Lambda$ defined in (6.5) must be non-singular. Unfortunately, we are unable to evaluate the true values of $\Lambda$ and $\gamma$, because they are very complicated complex functions of $\{s_m(t)\}_{m=1}^{P}$ and $\{n_k(t)\}_{k=1}^{M}$. In Theorem 6.4, we provide a convenient way to check whether $\Lambda$ is singular or not, when the signals are PM signals.

**Theorem 6.4** *Let $s_m(t)$, $\forall m$, be the PM signals described in **[AS2]**, i.e., $s_m(t)$, $\forall m$, have constant amplitudes and uniformly distributed phases. If $\Lambda$ is singular, then $\mathbf{C} = \gamma\mathbf{I}$, which is equivalent to the statement that if $\mathbf{C} \neq \gamma\mathbf{I}$, then $\Lambda$ is non-singular.*

**Proof**: Substituting $s_q(t) = u_q e^{j\psi_q(t)}$, $\forall q$, (see **[AS2]**) into (6.5), we have

$$\begin{aligned}
\Lambda_{mk} &= \delta_{mk}\, E\{u_m e^{j\psi_m(t)} | \sum_{q=1}^{P} u_q e^{j\psi_q(t)} + n_m(t)|^{p-2} (\sum_{r=1}^{P} u_r e^{j\psi_r(t)} + n_m(t))^*\} \\
&= \delta_{mk}\, u_m \underbrace{E\{e^{j\psi_m(t)} | \sum_{q=1}^{P} u_q e^{j\psi_q(t)} + n_m(t)|^{p-2} (\sum_{r=1}^{P} u_r e^{j\psi_r(t)} + n_m(t))^*\}}_{\beta} \\
&= \delta_{mk}\, u_m\, \beta \qquad (6.11)
\end{aligned}$$

The underbraced term in the second line of (6.11) is denoted $\beta$, and is a constant that is independent of indices, because $\psi_m(t)$ and $n_m(t)$, $\forall m$, are *i.i.d.* random variables (see **[AS2]** and **[AS3]**). Note that $\Lambda$ is a diagonal matrix, and the amplitudes of all signals, $u_m$, $\forall m$, are non-zero. We conclude, from (6.11), that if $\Lambda$ is singular, then $\beta = 0$ and all the diagonal entries of $\Lambda$ are zero, indicating $\Lambda$ is a zero matrix. Substituting $\Lambda = \mathbf{0}$ into (6.7), we obtain $\mathbf{C} = \gamma\mathbf{I}$.

The equivalent statement is obtained from simple logic. □

Theorem 6.4 is about the applicability of our FLOM-based MUSIC when the signals are PM. In practice, we use the equivalent statement in Theorem 6.4, that if $\mathbf{C} \neq \gamma\mathbf{I}$, then $\Lambda$ is non-singular. Note that $\mathbf{C}$ can be obtained from estimation, but that $\Lambda$ is unavailable; so, using Theorem 6.4, we can check whether to use MUSIC for $\mathbf{C}$ in (6.7): if $\mathbf{C} \neq \gamma\mathbf{I}$, use MUSIC; otherwise, do not use MUSIC.

We have not been able to arrive at a result that is comparable to the one in Theorem 6.4 when the signals are not PM. For such signals, we do not know a priori whether $\Lambda$ is singular or not.

In both cases (PM signals and non-PM signals), we apply singular value decomposition (SVD) to $\mathbf{C}$ to determine subspace information about $\mathbf{A}$. With minor modification to the work in [97], it is straightforward to show that MUSIC is applicable in this case for DOA estimation.

The well-known MUSIC algorithm has several implementations. The steps below are a noise-subspace-based MUSIC. The generalizations of our method to other MUSIC variants are straightforward. Our FLOM-based MUSIC (FLOM-MUSIC) using fractional moment $1 < p < \alpha$ is comprised of the following steps:

- Compute the $M \times M$ matrix $\widehat{\mathbf{C}}$, whose $(i,k)$-th entry is

$$\widehat{C}_{ik} = \frac{1}{N}\sum_{t=1}^{N} x_i(t)|x_k(t)|^{p-2}x_k^*(t). \tag{6.12}$$

  FLOM parameter $p$ is chosen by trial and error; some guidelines for doing this are given in Section 6.3.

- Perform SVD on $\widehat{\mathbf{C}}$ and construct the $M\times(M-P)$ matrix $\mathbf{E}_n \triangleq [\widehat{\mathbf{e}}_{P+1}, \widehat{\mathbf{e}}_{P+2}, \cdots, \widehat{\mathbf{e}}_M]$, where $\{\widehat{\mathbf{e}}_{P+1}, \widehat{\mathbf{e}}_{P+2}, \cdots, \widehat{\mathbf{e}}_M\}$ are the left singular vectors associated with the smallest $M-P$ singular values of $\widehat{\mathbf{C}}$.

- Compute the null spectrum

$$S(\phi) = \frac{1}{\mathbf{a}^H(\phi)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\phi)}, \qquad -90 \le \phi \le 90 \tag{6.13}$$

  where the $M \times 1$ steering vector $\mathbf{a}(\phi)$ is

$$\mathbf{a}(\phi) = [1 \quad e^{-j2\pi\frac{d_2}{\lambda}\sin\phi} \quad e^{-j2\pi\frac{d_3}{\lambda}\sin\phi} \quad \cdots \quad e^{-j2\pi\frac{d_M}{\lambda}\sin\phi}]^T \tag{6.14}$$

- Choose the $P$ local maxima of $S(\phi)$ as the estimates of DOAs, $\{\widehat{\phi}_1, \widehat{\phi}_2, \cdots, \widehat{\phi}_P\}$.

## 6.3 Simulations

Since an isotropic complex $S\alpha S$ random variable is characterized by two parameters, $\alpha$ and $\sigma$, we describe the signal-to-noise condition using $\alpha$ and a Generalized Signal-to-Noise Ratio (GSNR), where

$$\text{GSNR} \triangleq 10 \log \frac{E\{|s(t)|^2\}}{\sigma^\alpha} \tag{6.15}$$

If $\alpha = 2$, then the GSNR is the ratio of signal power to noise power, commonly known as the signal-to-noise ratio (SNR). See [69, 88] for the method to generate complex isotropic $S\alpha S$ sequences.

In our simulations, we assume that there are two signals ($\phi_1 = 0°$ and $\phi_2 = 8°$) that have the same power; therefore, in each experiment, we use an $\alpha$ and one GSNR to describe the signal-to-noise condition. The array under consideration is a 5-element uniform linear array (ULA) with interelement spacing equal to half a wavelength.

Two quantities are used to evaluate the performance of FLOM-MUSIC: success rate and mean-squared error (MSE) of the successful estimates. For each Monte-Carlo (MC) iteration, we define FLOM-MUSIC to be successful in resolving the two closely spaced signals if its null spectrum [see (6.13)] exhibits *exactly* two peaks inside the interval $[-4°, 12°]$, and, the locations of the two peaks are the successful estimates, $\widehat{\phi}_1$ and $\widehat{\phi}_2$. Success rate is the ratio, in percentage, of the number of successful MC iterations to the total number of MC iterations. MSE is the averaged sample mean-squared error of those successful estimates, i.e.,

$$\text{MSE} \triangleq \frac{1}{2K}\sum_{k=1}^{K}(\widehat{\phi}_1(k) - \phi_1)^2 + \frac{1}{2K}\sum_{k=1}^{K}(\widehat{\phi}_2(k) - \phi_2)^2 \tag{6.16}$$

where $\widehat{\phi}_1(k)$ and $\widehat{\phi}_2(k)$ are the respective successful estimates of $\phi_1$ and $\phi_2$ obtained from the $k$-th MC iteration, and $K$ is the total number of successful MC iterations.

Section 6.3.1 investigates the performance of our FLOM-MUSIC for the scenario when the PM signals (see **[AS2]**) are embedded in $S\alpha S$ noise. Section 6.3.2 shows that our FLOM-MUSIC can also be applied to the scenario that contains circularly

symmetrical Gaussian signals. Section 6.3.3 demonstrates that FLOM-MUSIC can also be applied for the scenario where the additive noise is no longer impulsive.

### 6.3.1 Experiment 1: $S\alpha S$ Noise

We performed MC simulations for scenarios where slightly ($\alpha = 1.8$), moderate ($\alpha = 1.5$), and highly ($\alpha = 1.2$) impulsive noises are present. The two signals are PM signals.

For each MC iteration, 1000 data snapshots were generated using a fixed $\alpha$ and GSNR; FLOM-MUSIC was then applied to the same data using different moments, $p = 1.0, 1.1, \cdots, 2.0$; a subtotal of 11 cases of FLOM-MUSIC were performed. We ran 500 MC iterations for each GSNR = 20, 15, 10, and 5 dB and a fixed $\alpha$; therefore, a total of $4 \times 11 \times 500 = 22,000$ complete calculations of FLOM-MUSIC were performed for each fixed $\alpha$ in this experiment. Three values of $\alpha$ (1.8, 1.5, and 1.2) were used; so, in totality, 66,000 calculations of FLOM-MUSIC were performed. Although the FLOM-based matrix is unbounded for $\alpha < p \leq 2$ (see Theorem 6.1), we still conducted FLOM-MUSIC using $\alpha < p \leq 2$, as well as for $1 < p \leq \alpha$. Success rates and MSEs versus FLOM are plotted in Fig. 6.1.

From Figs. 6.1(a) and (b), we observe that for all GSNR conditions, FLOM-MUSIC using $1 \leq p \leq 1.3 \leq \alpha$ gives better results, in terms of higher success rate and lower MSE, than FLOM-MUSIC using other moments. This observation indicates that the fractional moment $p$ used by FLOM-MUSIC should be selected smaller than $\alpha$ to yield good performance. Similar observations and conclusions can be drawn from Figs. 6.1(c)-(f). Note that in Figs. 6.1(d) and (f), the fact that the MSEs for GSNR = 5 dB and $p$ close to 2 are fairly small does not indicate FLOM-MUSIC using these $p$'s yields good performance, because the corresponding success rates are also fairly low in Figs. 6.1(c) and (e), respectively. Consequently, FLOM $p$ for the three scenarios, $\alpha = 1.8$, 1.5, 1.2 should be selected as $p < 1.3$, close to 1, and close to 1, respectively, to yield good performance.

Another interesting observation is that the estimates given by FLOM-MUSIC using $\alpha = 1.8 \leq p \leq 2.0$ do not necessarily converge in the mean-squared sense. To see this, we depict, in Fig. 6.2, the running MSEs of $\widehat{\phi}_1$, given by FLOM-MUSIC using $p = 2.0, 1.8, 1.4, 1.2$, when $\alpha = 1.8$ and GSNR = 20, 15, 10, 5 dB. The running

MSE versus the number of MC iterations, based on successful estimation, is defined as

$$\text{MSE}(K) \triangleq \frac{1}{K}\sum_{k=1}^{K}(\widehat{\phi}_1(k) - \phi_1)^2, \qquad K = 1, 2, 3, \cdots \tag{6.17}$$

Because of different success rates [see Figs.6.1(a), (c), and (e)], the curves in Fig.6.2 are of different lengths in abscissas.

Observe that the running MSE curves of the estimates given by FLOM-MUSIC using $p = 1.2$ and 1.4 are roughly flat after $K = 50$, and that those curves corresponding to FLOM-MUSIC using $p = 1.8$ and 2.0 are very jagged and seem to grow (although not monotonically). This observation indicates that FLOM-MUSIC using $p = 1.2$ or 1.4 is a convergent estimator for the scenario, and FLOM-MUSIC using $p = 1.8$ or 2.0 is not. This observation is not surprising, since in Theorem 6.1, FLOM-based matrices with $p = 1.8$ and 2.0 are unbounded for $S\alpha S$ noise with $\alpha = 1.8$.

### 6.3.2 Experiment 2: Two Examples of Circular Signals

In this experiment, we repeat the MC simulations for two different scenarios. Scenario #1, as the one in Section 6.3.1, is for the two PM signals embedded in additive $S\alpha S$ noise. Scenario #2 is for two circularly symmetrical Gaussian signals embedded in additive $S\alpha S$ noise; it is obtained by replacing the PM signals in scenario #1 with Gaussian signals. Note that circularly symmetrical Gaussian signals have Rayleigh-distributed amplitudes and uniformly distributed phases [59, 60]. We assume that these signals are zero-mean and circularly symmetrical with covariance $\eta\mathbf{I}$, so that they have the same power. The GSNR for scenario #2 is defined as $10\log(\eta/\sigma^2)$. The objective of this experiment is to show that our FLOM-MUSIC is not limited to PM signals.

MC simulations of our FLOM-MUSIC were applied to the two scenarios. For each MC simulation of scenario #1, we generated 1000 samples of the PM signals and $S\alpha S$ noise, which were then combined, according to (6.1), to provide its measurements. For each MC simulation of scenario #2, we generated 1000 samples of Gaussian signals, which were then combined with the noise component of scenario

#1 to provide its measurements. Hence, the two scenarios shared the same noise components for each MC run. A total of 500 MC runs were performed for the two scenarios where $\alpha = 1.8, 1.5$, and 1.2, and, GSNR = 20, 15, 10, and 5 dB. Note that the simulations for scenario #1 are exactly the same as in Experiment 1.

Simulation results are depicted in Fig. 6.3. Since scenario #1 is the same as Experiment 1, MC simulation results should be approximately the same for these two cases. Observe that the solid curves in Fig. 6.3 agree with the solid curves in Fig. 6.1. Figure 6.3 demonstrates that the solid (scenario #1) and dashed curves (scenario #2) are very close in all cases; hence, both PM and Gaussian signals yield similar performance. Observe, as in the case of Experiment 1, that FLOM-MUSIC yields best performance for both scenarios when the FLOM p is selected close to unity.

### 6.3.3 Experiment 3: Gaussian Noise

We repeated the same experiment as described in Section 6.3.1 for additive Gaussian noise ($S\alpha S$ noise with $\alpha = 2$). The objective of this experiment is to investigate how FLOM-MUSIC works when the additive noise is no longer impulsive. Success rates and MSEs versus FLOM are plotted in Figs. 6.4(a) and (b).

The roughly flat curves in Figs. 6.4(a) and (b) for $p \geq 1.5$ reveal that the use of FLOMs with $p = 1.5, 1.6, \cdots, 1.9$ does not degrade performance very much, as compared with the use of correlations (i.e., FLOM with $p = 2$). Still, the use of correlations leads to the lowest MSEs. This fact indicates that when the noise is Gaussian, correlation-based MUSIC is still a better candidate over FLOM-based MUSIC.

FLOM-MUSIC using $p \geq 1.5$ slightly outperforms FLOM-MUSIC that uses $p \leq 1.4$ for moderate noise situations (GSNR = 10, 15, 20 dB), as shown by the same success rate of 100% but slightly larger MSEs. For the poor noise condition (GSNR = 5 dB), FLOM-MUSIC using $p \geq 1.5$ outperforms, in terms of higher success rate, FLOM-MUSIC that uses $p \leq 1.4$, because the moment-based estimator of (6.12) degrades as the FLOM $p$ decreases [88]. One way to improve performance when $p$ is small is to use more data snapshots.

We conclude that when the additive noise is Gaussian, the use of FLOMs with $1.5 < p < 2.0$ does not degrade performance very much, as compared with the use of correlations (FLOM with $p = 2$).

## 6.4 Discussions

The class of FLOM-based matrices $\mathbf{C}$ defined in Theorem 6.1 is not the only one that can be processed by MUSIC to retrieve DOAs. Observe, from Theorem 6.1 and (6.1), that $\mathbf{C}$ retains the subspace information because its $(i,k)$-th entry $C_{ik}$ is the cross correlation of $x_i(t)$ (which is a linear combination of the signals) and a FLOM-based function of $x_k(t)$ (which is used to reduce the effects of impulsive noise).

We can define a second class of FLOM-based matrices, $\Gamma$, whose $(i,k)$-th entry is

$$\Gamma_{ik} \triangleq \frac{E\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}}{E\{|x_k(t)|^p\}}, \qquad 1 < p \leq 2 \tag{6.18}$$

Following the same procedure as we did in Section 6.2 for $\mathbf{C}$, we can prove that $\Gamma$ can be used with MUSIC to yield DOA estimates. Actually, $\Gamma$ was originally proposed by [88] using covariations and FLOMs, and the use of $\Gamma$ with MUSIC is called ROC-MUSIC. It is easy to see that $\mathbf{C}$ and $\Gamma$ are related by

$$\mathbf{C} = \Gamma \; diag\{E\{|x_1(t)|^p\}, E\{|x_2(t)|^p\}, \cdots, E\{|x_M(t)|^p\}\} \tag{6.19}$$

From the fact in Theorem 6.1 that $C_{kk} \triangleq E\{|x_k(t)|^p\}$, $\forall k$, and Theorem 6.3, we know that the diagonal matrix on the RHS of (6.19) is a multiple of the identity matrix, and (6.19) becomes

$$\mathbf{C} = E\{|x_k(t)|^p\}\,\Gamma, \qquad \forall k \tag{6.20}$$

i.e., $\mathbf{C}$ is a scaled version of $\Gamma$.

A third class of FLOM-based matrices $\Omega$ can be defined as

$$\Omega_{ik} \triangleq \frac{E\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}}{E\{|x_k(t)|^q\}}, \qquad 1 < p \leq 2, \quad 1 < q \leq 2. \tag{6.21}$$

Obviously, $\Omega$ is also a scaled version of $\mathbf{C}$, i.e.,

$$\mathbf{C} = E\{|x_k(t)|^q\}\,\Omega, \qquad \forall k \tag{6.22}$$

Continuing in this way, we can define many FLOM-based matrices that can be used with MUSIC to give DOA estimates. An interesting but open question is: how to find a FLOM-based matrix that yields the best performance.

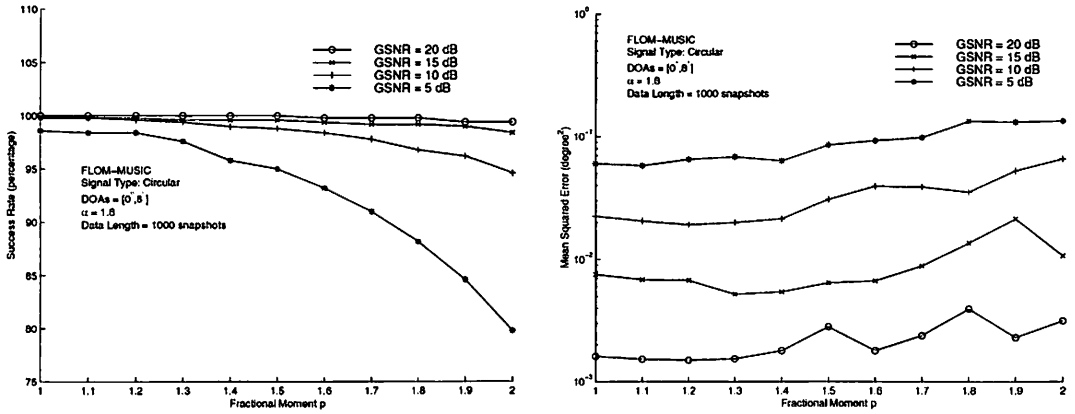
The combination of assumptions **[AS2]** and **[AS3]** is equivalent to the fact that the joint distribution of array outputs is available. In general, if the joint PDF of measurements is available, the derivation of optimal estimators of DOAs using maximum likelihood (ML) [83] is possible; however, ML estimators are always computationally intensive and usually do not have closed-form expressions. On the other hand, our suboptimal FLOM-MUSIC only requires a one-dimensional search over the interval $[-90°, 90°]$. Our contribution is, therefore, to propose a FLOM-based MUSIC that requires low computational costs to extract DOAs when the additive noise is $S\alpha S$. Recall that MUSIC-like methods have good statistical properties [73].

In implementation, we have to determine FLOM $p$ before conducting FLOM-MUSIC. The values for $p$ that give good performance are: $p$ close to 1 when the noise is $S\alpha S$ (Sections 6.3.1 and 6.3.2), and $1.5 < p < 2.0$ when the noise is Gaussian (Section 6.3.3). It is best to know the parameter $\alpha$ of the $S\alpha S$ noise; otherwise, we have to estimate it using a method such as the one in [82], so that we can choose the appropriate value for $p$.

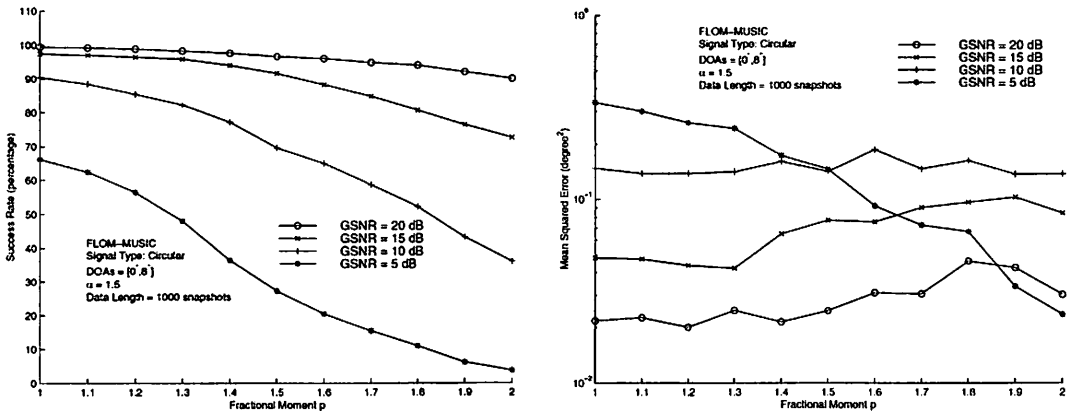
There are several similarities and differences between our work and that of [88]. First, both [88] and our work proposed a class of matrices that can be used with MUSIC to extract DOAs; [88] used covariations and FLOMs, and we only used FLOMs. Second, [88] assumed that the signals are jointly $S\alpha S$ with the additive noise, which may be difficult or impossible to justify in practical applications, whereas we assumed that the signals are circular and independent of the additive $S\alpha S$ noise. Third, estimators for the proposed matrices are both based on FLOMs. Fourth, [88] performed eigenvalue decomposition on $\Gamma + \Gamma^H$, whereas we performed SVD on $\mathbf{C}$ to retrieve subspace information from the non-Hermitian $\mathbf{C}$. Fifth, we generalized our FLOM-MUSIC to propose several classes of matrices; there may exist more FLOM-based matrices that can be used with MUSIC.

## 6.5 Conclusions

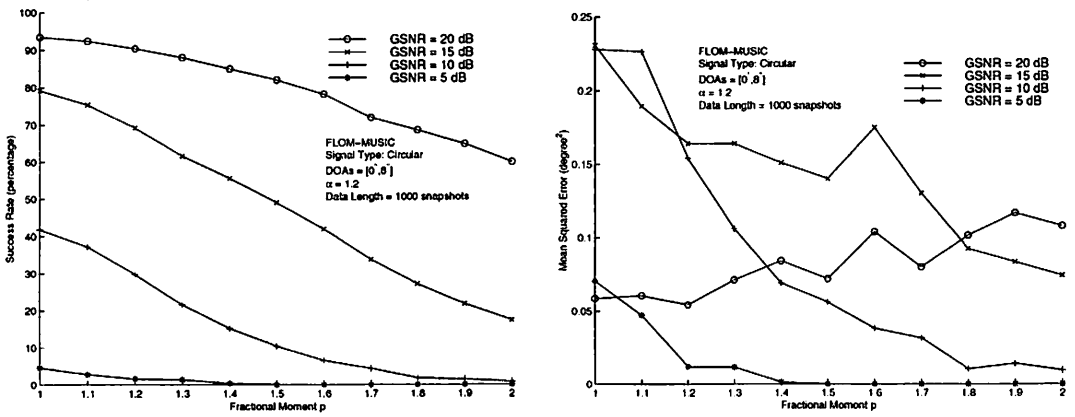
We have proposed several classes of FLOM-based matrices that can be used with MUSIC to estimate the DOAs of independent circular signals embedded in additive $S\alpha S$ noise. When the signals have constant amplitudes and uniformly distributed phase, our FLOM-MUSIC works if $\mathbf{C} \neq \gamma\mathbf{I}$ (see Theorem 6.4); otherwise, the number of signals must be known. We ran simulations for different choices of $p$'s, and concluded that when the noise is $S\alpha S$ with $\alpha \neq 2$, the FLOM $p$ should be selected close to unity. Two examples of circular signals (PM and circularly symmetrical Gaussian signals) embedded in the same $S\alpha S$ noise were tested in Section 6.3.2; the simulation results revealed similar performance for the two scenarios when the signal-to-noise situation are the same. Our FLOM-MUSIC ($p > 1.5$) can also apply for the scenario where the additive noise is Gaussian without much performance degradation, when compared to correlation-based MUSIC.




(a) Success Rate ($\alpha = 1.8$). (b) Mean Squared Error ($\alpha = 1.8$).

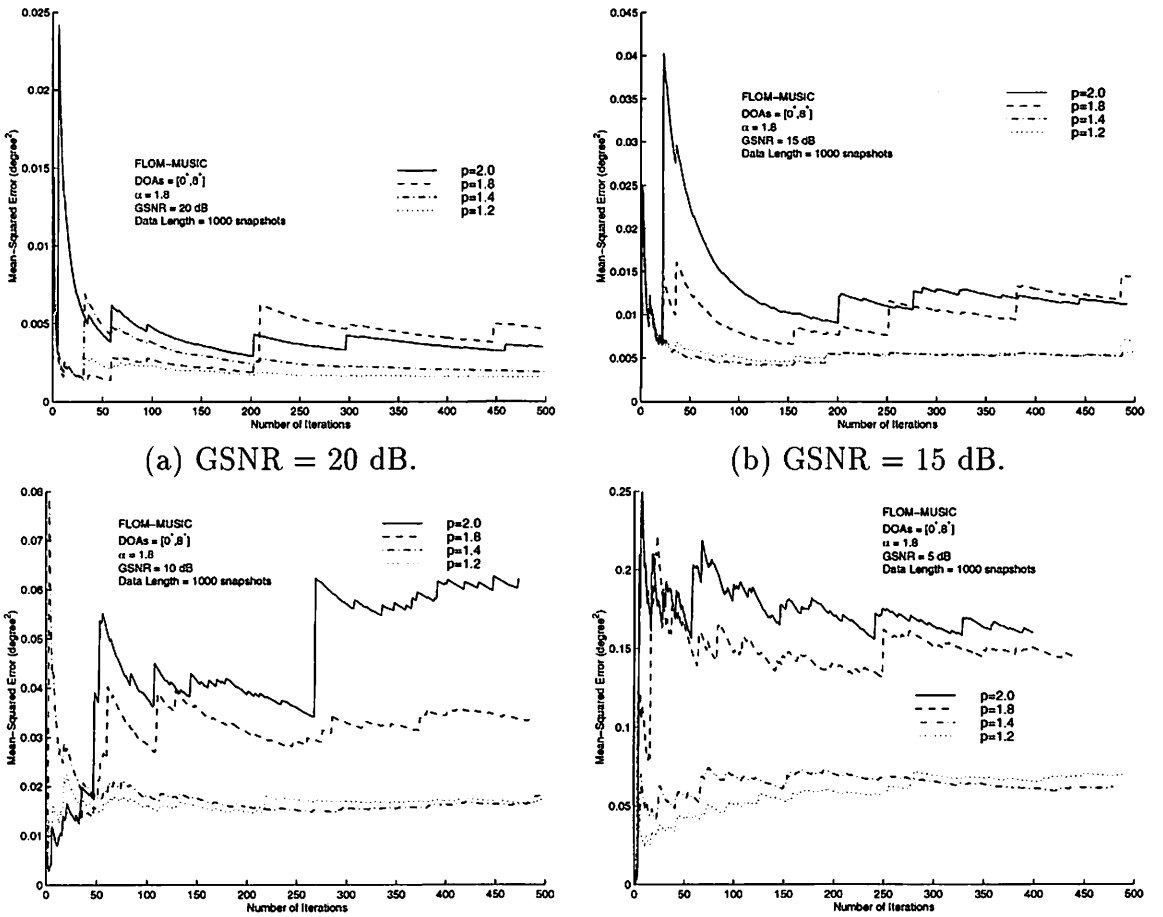



(c) Success Rate ($\alpha = 1.5$). (d) Mean Squared Error ($\alpha = 1.5$).




(e) Success Rate ($\alpha = 1.2$). (f) Mean Squared Error ($\alpha = 1.2$).

Figure 6.1: Success rates of FLOM-MUSIC in the presence of $S\alpha S$ noise: (a) $\alpha = 1.8$, (c) $\alpha = 1.5$, and (e) $\alpha = 1.2$. The corresponding mean-squared errors are depicted in (b), (d), and (f), respectively.



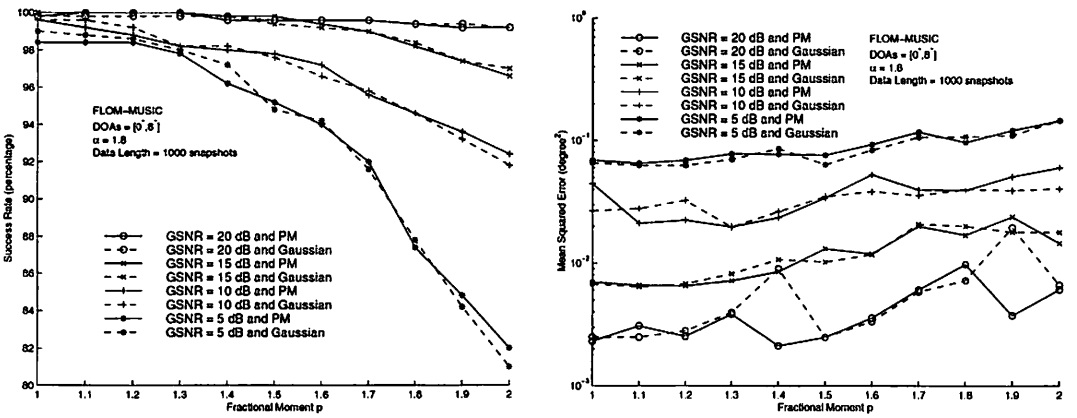

(a) GSNR = 20 dB.

(b) GSNR = 15 dB.

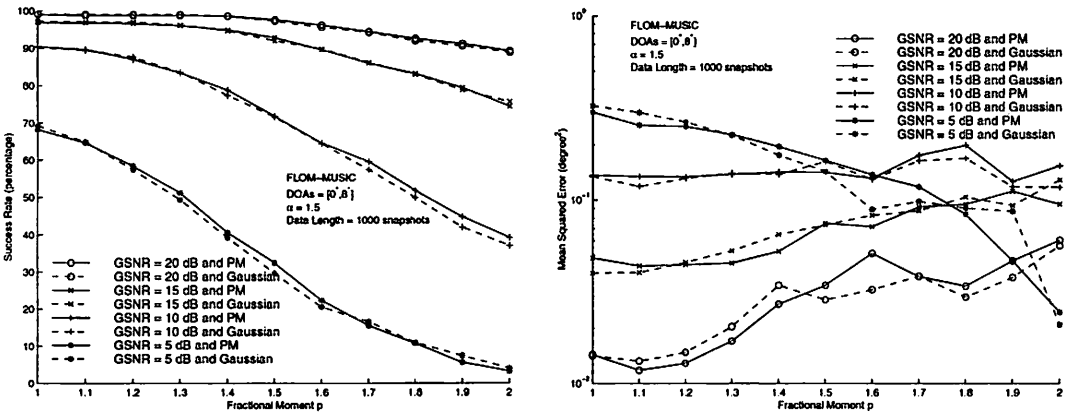
(c) GSNR = 10 dB.

(d) GSNR = 5 dB.

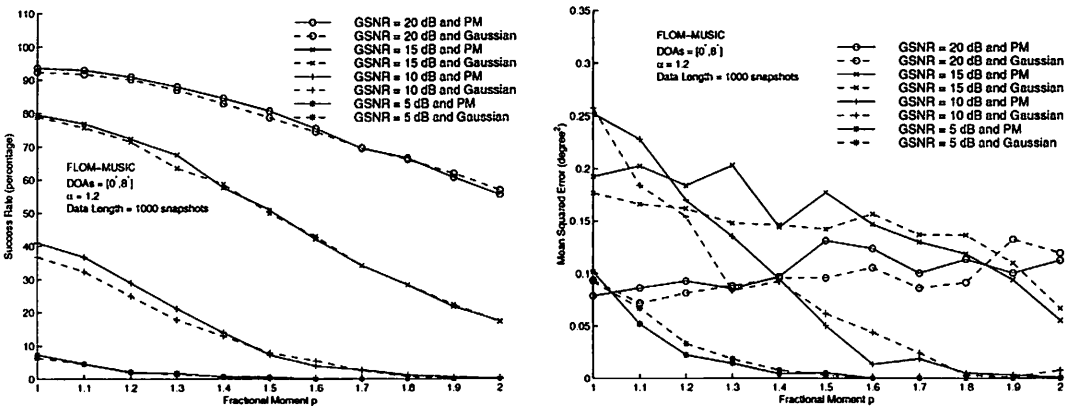
Figure 6.2: Running MSEs of the estimates of $\phi_1$ from FLOM-MUSIC versus number of Monte-Carlo iterations: (a) GSNR = 20 dB, (b) GSNR = 15 dB, (c) GSNR = 10 dB, and (d) GSNR = 5 dB.




(a) Success Rate ($\alpha = 1.8$). (b) Mean Squared Error ($\alpha = 1.8$).

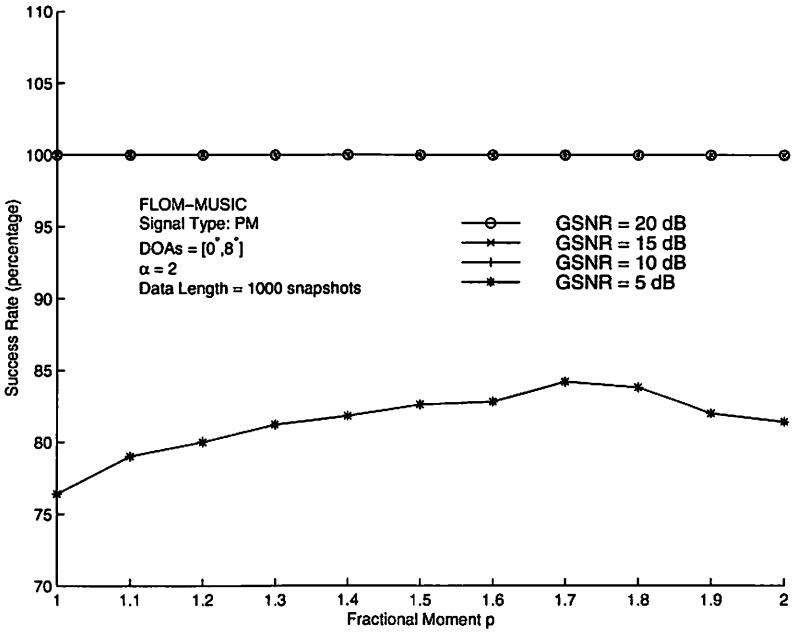



(c) Success Rate ($\alpha = 1.5$). (d) Mean Squared Error ($\alpha = 1.5$).

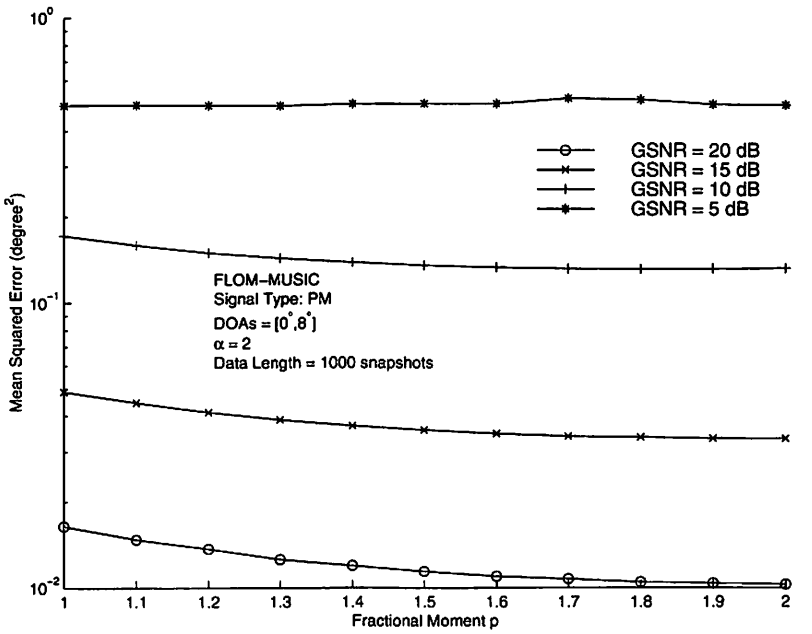



(e) Success Rate ($\alpha = 1.2$). (f) Mean Squared Error ($\alpha = 1.2$).

Figure 6.3: Success rates of FLOM-MUSIC in the presence of $S\alpha S$ noise: (a) $\alpha = 1.8$, (c) $\alpha = 1.5$, and (e) $\alpha = 1.2$. The corresponding mean-squared errors are depicted in (b), (d), and (f), respectively.




(a) Success Rate




(b) Mean Squared Error

Figure 6.4: Performance of FLOM-MUSIC in the presence of Gaussian noise: (a) success rate and (b) mean-squared error. Note that in (a), the success rates for GSNR = 10, 15, 20 dB are the same 100%.

# Chapter 7

# Subspace-Based Direction Finding Using Fractional Lower-Order Moments: ESPRIT

In Chapter 6, we applied fractional lower-order moments (FLOMs) to the scenarios where the signals are circular and the additive noise is symmetric alpha stable ($S\alpha S$) with $1 < \alpha \leq 2$. We showed that the resulting FLOM-based matrices can be used with MUSIC to retrieve the DOAs of signals. In this chapter, assuming that two identical subarrays are available, we propose a class of FLOM-based matrices that can be used with ESPRIT to extract DOAs. The motivation of using ESPRIT, rather than MUSIC, is computational. Unlike MUSIC that needs a searching procedure, ESPRIT algebraically calculates the DOAs, and, therefore, saves computational costs. Accordingly, ESPRIT trades higher hardware costs, due to the requirement of two identical subarrays, for less computational costs.

In Section 7.1, we define the problem of interest. In Section 7.2, we propose a FLOM-based ESPRIT to extract DOAs of the signals. Simulations are presented in Section 7.3. Finally, we give discussions and draw conclusions in Sections 7.4 and 7.5, respectively.

## 7.1 Problem Definition

Assume that there are $P$ narrowband independent signals impinging on two identical subarrays of $M$ elements each. Array outputs at the $k$-th sensors of the two identical subarrays, in complex envelope representation, are

$$x_k(t) = \sum_{m=1}^{P} A_{km} s_m(t) + n_{x_k}(t), \qquad k = 1, 2, \cdots, M \tag{7.1a}$$

$$y_k(t) = \sum_{m=1}^{P} A_{km} e^{-j2\pi\Delta \sin\phi_m/\lambda} s_m(t) + n_{y_k}(t), \qquad k = 1, 2, \cdots, M \tag{7.1b}$$

where $A_{km}$ is the sensor response of the sensor $x_k$ with respect to the $m$-th signal, $\lambda$ is the wavelength of the carrier, $s_m(t)$ is the $m$-th signal measured at the reference point, $\phi_m$ is the DOA of $s_m(t)$, $\Delta$ is the displacement between the two subarrays, and, $n_{x_k}(t)$ and $n_{y_k}(t)$ are the respective noises at sensors $x_k$ and $y_k$. The requirement of two identical subarrays leads to the fact that the sensor responses of $x_k$ and $y_k$ with respect to signal $s_m(t)$ are $A_{km}$ and $A_{km}e^{-j2\pi\Delta \sin\phi_m/\lambda}$, respectively. The matrix form for the entire array, from (7.1), is

$$\mathbf{z}(t) \triangleq \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{A}\Phi \end{bmatrix}}_{\bar{\mathbf{A}}} \mathbf{s}(t) + \underbrace{\begin{bmatrix} \mathbf{n}_x(t) \\ \mathbf{n}_y(t) \end{bmatrix}}_{\mathbf{n}_z(t)} = \bar{\mathbf{A}}\mathbf{s}(t) + \mathbf{n}_z(t) \tag{7.2}$$

where

| | |
|---|---|
| $\mathbf{z}_{2M\times 1}(t) \triangleq [x_1(t), \cdots, x_M(t), y_1(t), \cdots, y_M(t)]^T$ | measurement vector, |
| $\mathbf{A}_{M\times P} \triangleq \{A_{km}\}$ | array manifold, |
| $\mathbf{s}_{P\times 1}(t) \triangleq [s_1(t), \cdots, s_P(t)]^T$ | signal vector, |
| $\mathbf{n}_{z,2M\times 1}(t) \triangleq [n_{x_1}(t), \cdots, n_{x_M}(t), n_{y_1}(t), \cdots, n_{y_M}(t)]^T$ | noise vector, |

and

$$\Phi_{P\times P} \triangleq \text{diag}\{e^{-j2\pi\Delta \sin\phi_1/\lambda}, \cdots, e^{-j2\pi\Delta \sin\phi_P/\lambda}\} \tag{7.3}$$

Note that in (7.2), $\mathbf{z}(t)$ is the entire array, $\mathbf{x}(t)$ is the main subarray, and $\mathbf{y}(t)$ is the copy subarray. In later sections of this chapter, we will use $z_i(t)$, and, $x_i(t)$ and $y_i(t)$, $\forall i$, interchangeably.

The following assumptions are made throughout this chapter:

**[AS1]** $A_{km} \triangleq e^{-j\tau_{km}}$ and $\tau_{km} = 2\pi \frac{d_k}{\lambda} \sin(\phi_m)$, where $d_k$ is the distance between sensor $x_k$ and the reference point. This is a reasonable assumption for a calibrated array, where the sensor gains are known through calibration and are assumed to be unity. Note that the assumption of identical subarrays made in ESPRIT is achieved through calibration.

**[AS2]** $s_m(t) = u_m(t)e^{j\psi_m(t)}$, where amplitude $u_m(t)$ and phase $\psi_m(t)$ are real random processes that are statistically independent of each other, and $\psi_m(t)$ is a sequence of independent identically distributed (*i.i.d.*) random variables that are uniform over $[0, 2\pi)$. If $u_m(t)$ is a constant, then $s_m(t)$ is referred to as a phase modulated (PM) signal. If $u_m(t)$ is Rayleigh-distributed, then $s_m(t)$ is referred to as a circularly symmetric Gaussian signal. Both PM and circularly symmetric Gaussian signals belong to the class of circular signals. Note that $s_m(t)$ is zero-mean.

**[AS3]** $n_{x_k}(t)$ and $n_{y_k}(t)$ are sequences of *i.i.d.* isotropic complex $S\alpha S$ random variables with $1 < \alpha \leq 2$.

**[AS4]** Array manifold $\bar{\mathbf{A}}$ is of full rank.

Note that these assumptions are the same as those in Chapter 6, and that the main subarray $\mathbf{x}(t)$ is modeled the same way as the array in FLOM-MUSIC. The problem of DOA estimation is to extract the DOAs, $\{\phi_m\}_{m=1}^{P}$, from measurements $\{\mathbf{z}(t)\}_{t=1}^{N}$.

## 7.2 Proposed Solution

Our steps to propose a FLOM-based ESPRIT algorithm are similar to what we did for FLOM-MUSIC in Chapter 6; hence, we rely heavily on detailed derivations in Chapter 6. First, we propose a class of FLOM-based matrices that are bounded;

second, we express the proposed matrices in a form that can be used with ESPRIT to extract DOAs.

In Theorem 7.1, we define a class of bounded matrices.

**Theorem 7.1** *Let* $\mathbf{C}$ *be a* $2M \times 2M$ *matrix, whose* $(i,k)$*-th entry* $C_{ik}$ *is defined as*

$$C_{ik} \triangleq E\{z_i(t)|z_k(t)|^{p-2}z_k^*(t)\}, \qquad 1 < p < \alpha \leq 2 \tag{7.4}$$

*where* $z_i(t)$ *and* $z_k(t)$, $\forall i$ *and* $k$, *are the array outputs defined in Section 7.1. The* $C_{ik}$, $\forall i$ *and* $k$, *are bounded.*

**Proof**: Matrix $\mathbf{C}$ can be divided into four $M \times M$ submatrices,

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix} \tag{7.5}$$

where the $(i,k)$-th entries of $\mathbf{C}_{xx}$, $\mathbf{C}_{xy}$, $\mathbf{C}_{yx}$, and $\mathbf{C}_{yy}$ are $E\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}$, $E\{x_i(t)|y_k(t)|^{p-2}y_k^*(t)\}$, $E\{y_i(t)|x_k(t)|^{p-2}x_k^*(t)\}$, and $E\{y_i(t)|y_k(t)|^{p-2}y_k^*(t)\}$, respectively.

Although in Theorem 6.1 of Chapter 6 we only proved that entries of $\mathbf{C}_{xx}$ are bounded, the generalization to prove that all entries of $\mathbf{C}_{xy}$, $\mathbf{C}_{yx}$, and $\mathbf{C}_{yy}$ are bounded is similar to the proof for $\mathbf{C}_{xx}$, and is therefore omitted here; hence, all entries of $\mathbf{C}$ are bounded. □

Theorem 7.1 not only recommends FLOMs but also negates correlations, because the correlation of $z_i(t)$ and $z_k(t)$, which is a special case of $C_{ik}$ when $p = 2$, is unbounded. Theorem 7.1 is needed for analysis purposes, because only bounded quantities are meaningful in statistical analyses.

Next, we show, in Theorem 7.2, that $\mathbf{C}$ is of a particular structure that is useful for DOA estimation.

**Theorem 7.2** *Matrix* $\mathbf{C}$, *defined in Theorem 7.1, can be written as*

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\Phi \end{bmatrix} \Lambda \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\Phi \end{bmatrix}^H + \gamma\,\mathbf{I} \triangleq \bar{\mathbf{A}}\Lambda\bar{\mathbf{A}}^H + \gamma\,\mathbf{I} \tag{7.6}$$

*where the $(m,k)$-th entry of $\Lambda$ is*

$$\Lambda_{mk} \;=\; \delta_{mk}\, E\{s_m(t)|\sum_{q=1}^{P} s_q(t)+n_m(t)|^{p-2}(\sum_{r=1}^{P} s_r(t)+n_m(t))^*\} \qquad (7.7)$$

*and*

$$\gamma \;=\; E\{n_k(t)|\sum_{q=1}^{P} s_q(t)+n_k(t)|^{p-2}(\sum_{r=1}^{P} s_r(t)+n_k(t))^*\}, \qquad (7.8)$$

*in which $\delta_{mk}$ is the Kronecker delta.*

**Proof**: We calculate independently the four $M \times M$ submatrices of $\mathbf{C}$ [see (7.5)], and then substitute them back into $\mathbf{C}$. Matrix $\mathbf{C}_{xx}$ is exactly the same as matrix $\mathbf{C}$ of Theorem 6.2 in Chapter 6, i.e.,

$$\mathbf{C}_{xx} \;\triangleq\; \mathbf{A}\Lambda\mathbf{A}^H + \gamma\mathbf{I} \qquad (7.9)$$

To compute $\mathbf{C}_{yx}$, observe that its $(i,k)$-th entry is

$$\begin{aligned} C_{yx}(i,k) \;&\triangleq\; E\{y_i(t)|x_k(t)|^{p-2}x_k^*(t)\} \\ &=\; E\{(\sum_{m=1}^{P} A_{im}\Phi_{mm}s_m(t)+n_{y_i}(t))|\sum_{q=1}^{P} A_{kq}s_q(t)+n_{x_k}(t)|^{p-2} \\ &\qquad\qquad (\sum_{r=1}^{P} A_{kr}s_r(t)+n_{x_k}(t))^*\} \\ &= \sum_{m=1}^{P} A_{im}\Phi_{mm} \underbrace{E\{s_m(t)|\sum_{q=1}^{P} A_{kq}s_q(t)+n_{x_k}(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}s_r(t)+n_{x_k}(t))^*\}}_{A_{km}^*\Lambda_{mm}} \\ &+ \underbrace{E\{n_{y_i}(t)|\sum_{q=1}^{P} A_{kq}s_q(t)+n_{x_k}(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}s_r(t)+n_{x_k}(t))^*\}}_{0} \end{aligned} \qquad (7.10)$$

The first underbraced term of (7.10) is equal to $A_{km}^*\Lambda_{mm}$, which is obtained by applying Theorem C.1 in Appendix C with $e^{-j\tau_{kq}} = A_{kq}$ (see **[AS1]**). The second term of (7.10) is zero, because $n_{y_i}(t)$, by **[AS3]**, is zero-mean and is independent of

all the other processes. We write (7.10) as $C_{yx}(i,k) = \sum_{m=1}^{P} A_{im}\Phi_{mm}\Lambda_{mm}A_{km}^*$, and its matrix form as

$$\mathbf{C}_{yx} = \mathbf{A}\Phi\Lambda\mathbf{A}^H \tag{7.11}$$

The computation of $C_{xy}(i,k)$ is so similar to that of $C_{yx}(i,k)$ that we omit its details, and merely state that

$$\mathbf{C}_{xy} = \mathbf{A}\Lambda\Phi^H\mathbf{A}^H \tag{7.12}$$

Finally, we compute $\mathbf{C}_{yy}$, whose $(i,k)$-th entry is

$$\begin{aligned}
C_{yy}(i,k) &\triangleq E\{y_i(t)|y_k(t)|^{p-2}y_k^*(t)\} \\
&= E\{\sum_{m=1}^{P} A_{im}\Phi_{mm}s_m(t) + n_{y_i}(t)|\sum_{q=1}^{P} A_{kq}\Phi_{qq}s_q(t) + n_{y_k}(t)|^{p-2} \\
&\qquad (\sum_{r=1}^{P} A_{kr}\Phi_{rr}s_r(t) + n_{y_k}(t))^*\} \\
&= \sum_{m=1}^{P} A_{im}\Phi_{mm} \underbrace{E\{s_m(t)|\sum_{q=1}^{P} A_{kq}\Phi_{qq}s_q(t) + n_{y_k}(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}\Phi_{rr}s_r(t) + n_{y_k}(t))^*\}}_{A_{km}^*\Phi_{mm}^*\Lambda_{mm}} \\
&\quad + \underbrace{E\{n_{y_i}(t)|\sum_{q=1}^{P} A_{kq}s_q(t) + n_{y_k}(t)|^{p-2}(\sum_{r=1}^{P} A_{kr}\Phi_{rr}s_r(t) + n_{y_k}(t))^*\}}_{\gamma\,\delta_{ik}}
\end{aligned} \tag{7.13}$$

The first underbraced term of (7.13) is equal to $A_{km}^*\Phi_{mm}^*\Lambda_{mm}$, which is obtained by applying Theorem C.1 with $e^{-j\tau_{kq}} = A_{kq}\Phi_{qq}$. The second term of (7.13) is $\gamma\ \delta_{ik}$, which is also obtained by using Theorem C.1 with $e^{-j\tau_{kq}} = A_{kq}\Phi_{qq}$ for $q = 1, 2, \cdots, P$. We write (7.13) as

$$C_{yy}(i,k) = \sum_{m=1}^{P} A_{im}\Phi_{mm}\Lambda_{mm}\Phi_{mm}^*A_{km}^* + \gamma\,\delta_{ik} \tag{7.14}$$

so that the matrix form for $\mathbf{C}_{yy}$ is

$$\mathbf{C}_{yy} = \mathbf{A}\Phi\Lambda\Phi^H\mathbf{A}^H + \gamma\,\mathbf{I} \tag{7.15}$$

Substituting (7.9), (7.11), (7.12), and (7.15) into (7.6) with careful rearrangement, we have the first equality of (7.6). The second equality of (7.6) follows from the definition of $\bar{\mathbf{A}}$, which is in (7.2). □

Theorem 7.2 reveals that $\mathbf{C}$ includes the DOA information in the array manifold $\bar{\mathbf{A}}$. Note that the correlation matrix of the array outputs, when additive white Gaussian noise is present, is $\mathbf{R}_{zz} = \bar{\mathbf{A}}\mathbf{R}_s\bar{\mathbf{A}}^H + \sigma_n^2\mathbf{I}$ [67], where positive-definite $\mathbf{R}_s$ is the correlation matrix of signals and $\sigma_n^2$ is the variance of the white Gaussian noise. Comparing $\mathbf{R}_{zz} = \bar{\mathbf{A}}\mathbf{R}_s\bar{\mathbf{A}}^H + \sigma_n^2\mathbf{I}$ with $\mathbf{C}$ in (7.6), we can interpret $\Lambda$ as a fractional lower-order correlation matrix of the signals, and $\gamma$ as a fractional lower-order correlation of the additive noise level. Recall that the same interpretation was made in Chapter 6.

To apply ESPRIT to $\mathbf{C}$ in (7.4) for DOA estimation, $\Lambda$ must be non-singular. If the signals are PM, we use Theorem 6.4 of Chapter 6 to check whether $\Lambda$ is singular or not. For non-PM signals, we do not know a priori whether $\Lambda$ is singular or not.

The well-known ESPRIT algorithm has several implementations. The steps below are a total-least-squared implementation of ESPRIT (TLS-ESPRIT) [67]. The generalizations of our method to other ESPRIT variants are straightforward. Our FLOM-based ESPRIT (FLOM-ESPRIT) using fractional moment $1 < p < \alpha$ is comprised of the following steps:

- Compute the $2M \times 2M$ matrix $\widehat{\mathbf{C}}$, whose $(i,k)$-th entry is

$$\widehat{C}_{ik} = \frac{1}{N}\sum_{t=1}^{N} z_i(t)|z_k(t)|^{p-2}z_k^*(t). \tag{7.16}$$

  Note that the (fractional lower-order) moment-based estimator is used in (7.16). FLOM parameter $p$ is chosen by trial and error; some guidelines for doing this are given in Section 7.3.

- Perform SVD on $\mathbf{C}$; retain those left singular vectors associated with the $P$ largest singular values in $\mathbf{E}_s$, which is of dimension $2M \times P$; partition $\mathbf{E}_s$ into two $M \times P$ matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, i.e., $\mathbf{E}_s \triangleq [\mathbf{E}_1^T \ \mathbf{E}_2^T]^T$.

- Perform SVD on the $M \times 2P$ matrix $\mathbf{E}_{12} \triangleq [\mathbf{E}_1 \ \mathbf{E}_2]$; place the last $P$ right singular vectors of $\mathbf{E}_{12}$ in $\mathbf{V}_2$; partition $\mathbf{V}_2$ into two $P \times P$ matrices $\mathbf{V}_{12}$ and $\mathbf{V}_{22}$, i.e., $\mathbf{V}_2 \triangleq [\mathbf{V}_{12}^T \ \mathbf{V}_{22}^T]^T$.

- Perform eigenvalue decomposition on the $P \times P$ matrix $\Psi \triangleq -\mathbf{V}_{12}\mathbf{V}_{22}^{-1}$; let $\Phi_{pp}$, $p = 1, 2, \cdots, P$ be the eigenvalues.

- Compute $\hat{\phi}_p$ using

$$\hat{\phi}_p = \arcsin\left(\frac{\angle \Phi_{pp}}{2\pi\triangle}\right) \tag{7.17}$$

## 7.3 Simulations

In our simulations, we assume that there are two PM signals ($\phi_1 = 0°$ and $\phi_2 = 8°$) that have the same power; therefore, in each experiment, we use an $\alpha$ and a GSNR [see (6.15)] to describe the signal-to-noise condition. The array under consideration is a 10-element uniform linear array (ULA) with interelement spacing equal to a half wavelength. The odd-numbered sensors form the main subarray, and the even-numbered sensors form the second subarray. Accordingly, we have two identical subarrays of 5 elements each, and the displacement between the subarrays is a half wavelength.

Mean-squared error (MSE) [see (6.16)] is used to evaluate the performance of FLOM-ESPRIT.

We performed 500 Monte-Carlo (MC) iterations of FLOM-ESPRIT for scenarios where $\alpha = 2.0$, 1.8, 1.5, and 1.2 and GSNR = 20, 15, 10, 5, and 0 dB. For each MC run, 1000 snapshots were generated and FLOM-ESPRIT was then applied to the same data using different moments, $p = 1.0, 1.1, \cdots, 2.0$. The MSEs versus FLOM $p$ are plotted in Fig. 7.1. Note that when $p = 2$, our FLOM-ESPRIT becomes the correlation-based ESPRIT.

When the noise is Gaussian ($S\alpha S$ with $\alpha = 2$), the nearly flat curves for GSNR = 0 and 5 dB (poor signal-to-noise condition) in Fig. 7.1(a) reveal that the performance of FLOM-ESPRIT does not change much over FLOMs. For GSNR = 10, 15, and 20 dB (good signal-to-noise condition), MSEs of the estimates given by FLOM-ESPRIT increase as $p$ decreases [see Fig. 7.1(a)], revealing that correlation-based ESPRIT outperforms FLOM-ESPRIT. Basically, FLOM-ESPRIT still performs fairly well for this situation.

Figure 7.1(b) is for the slight ($\alpha = 1.8$) $S\alpha S$ situation. When GSNR is high (20 dB), FLOM-ESPRIT degrades as FLOM decreases, because FLOM-ESPRIT with $p = 2$ is still quite robust to such high signal-to-noise situation. As GSNR decreases, FLOM-ESPRIT with $1.0 < p < 1.5$ outperforms FLOM-ESPRIT with $1.5 < p < 2.0$. For GNR = 5 and 0 dB, FLOM-ESPRIT has lowest MSE when $p$ is close to unity.

When the $S\alpha S$ noise is large ($\alpha = 1.5$ or 1.2), FLOM-ESPRIT has lowest MSE when $p$ is close to unity, a fact observed from Figs. 7.1(c) and (d). Note that the curves that correspond to low GSNRs in Figs. 7.1(c) and (d) have large MSEs, indicating that FLOM-ESPRIT performs poorly for such noisy conditions.

We conclude that when the noise is medium or highly $S\alpha S$, FLOM-ESPRIT yields the lowest MSE when $p$ is close to unity. Note that we drew the same conclusion from the simulation results for FLOM-MUSIC Chapter 6. This conclusion is not surprising, since FLOM-MUSIC and FLOM-ESPRIT are applied to the similar matrix structure given by (7.9) and (7.6).

It is not fair to compare our MSE results for FLOM-ESPRIT and FLOM-MUSIC. In Chapter 6 the performance measures for FLOM-MUSIC are success rates and MSEs of only the successful estimates, whereas for FLOM-ESPRIT only MSE is used, assuming a 100% success rate.

## 7.4 Discussions

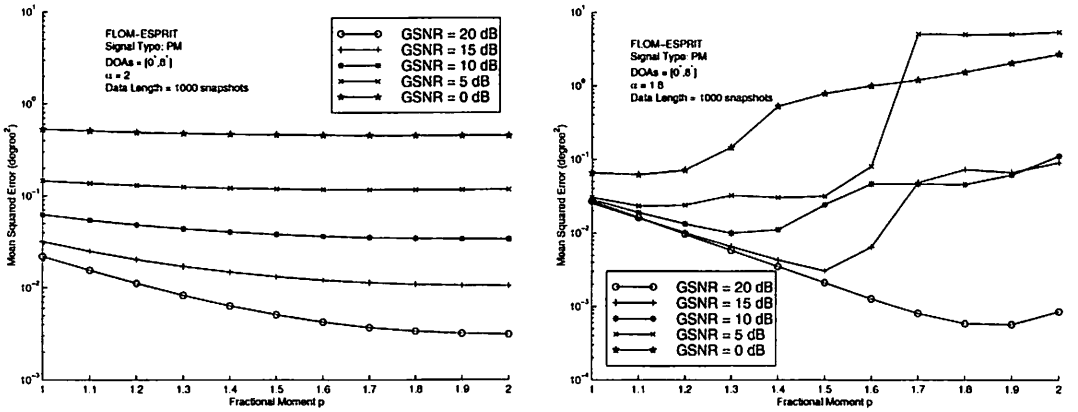
Because MUSIC and ESPRIT share similar properties, it is not surprising that FLOM-MUSIC and FLOM-ESPRIT share similar properties. Note that: (1) the diagonal entries of $\mathbf{C}$ are identical, because $\mathbf{C}_{xx}$ has identical diagonal entries according to Theorem 6.3 in Chapter 6, and $\mathbf{C}_{yy} = \mathbf{A}\Phi\Lambda\Phi^H\mathbf{A}^H = \mathbf{A}\Lambda\Phi\Phi^H\mathbf{A}^H =$

$\mathbf{A}\Lambda\mathbf{A}^H = \mathbf{C}_{xx}$ [see (7.15) and (7.9)] due to the facts that $\Phi$ and $\Lambda$ are diagonal matrices, and, $\Phi^H\Phi = \mathbf{I}$ [see (7.3)]; (2) based on Theorem 7.1 and the fact that the diagonal entries of $\mathbf{C}$ are identical, we can define more classes of bounded matrices as we did in Chapter 6; (3) both FLOM-MUSIC and FLOM-ESPRIT work if the $\Lambda$ in (7.6) is non-singular; and, (4) both algorithms yield best results when FLOM $p$ is close to unity.

The recently published Method of Reflection of Subspaces (MRS) [43] uses just the *cross correlations* of the outputs from two identical subarrays. MRS can remove all types of additive noise, including $S\alpha S$ noise, so long as the noise components are uncorrelated with one another, i.e., MRS can be applied to the scenario described in this chapter. Unfortunately, MRS is constrained by the fact that the displacement between the two identical subarrays must be less than a quarter wavelength. Generally, antennas require sufficient spacing between neighboring elements to radiate efficiently; sufficient spacing also makes the assumption of independent noise reasonable. Accordingly, the constraint of a quarter wavelength is impractical for array antennas.

Actually, our FLOM-MUSIC and FLOM-ESPRIT can be applied to other types of noise processes, in addition to $S\alpha S$ processes. Recall, in our derivations, that the probability density functions of $S\alpha S$ random processes are never used. The only property we used about $S\alpha S$ processes is that they have finite fractional moments [see (A.4)]. For this reason, our FLOM-MUSIC and FLOM-ESPRIT can be applied to any scenario whose noise components are of bounded FLOMs.

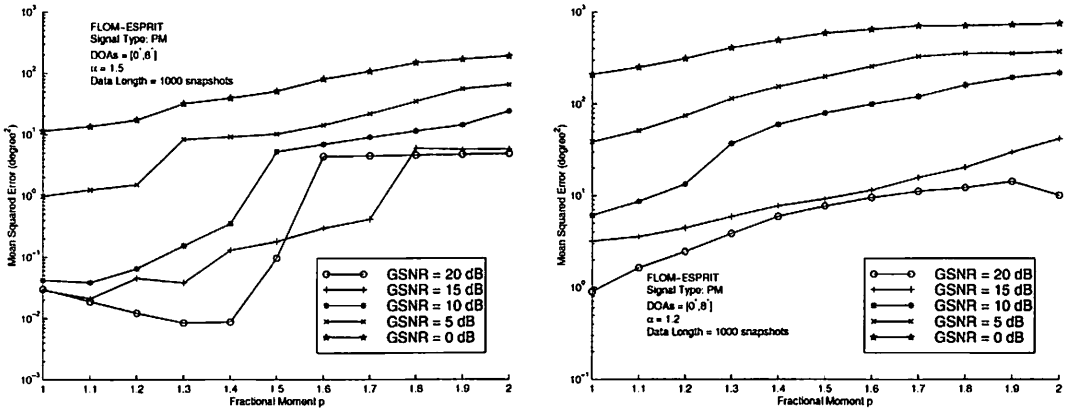
Note that $\mathbf{C}$ in (7.6) is not necessarily Hermitian symmetric, because we did not prove that $\mathbf{C}_{xy}^H = \mathbf{C}_{yx}$. We can prove that $\mathbf{C}_{xy}^H = \mathbf{C}_{yx}$, if the noise is circularly symmetrical. As we stated in the preceding paragraph, assumption of circular noise limits the applicability of our FLOM-MUISC and FLOM-ESPRIT. Matrix $\Lambda$ in (7.6) is not necessarily positive semi-definite; hence, eigenvalue decomposition (EVD) cannot be applied to $\mathbf{C}$ for DOA estimation. A similar situation occurs in [29] when applying cumulants to the array outputs, where the resulting cumulant-based matrix is not guaranteed to be positive semidefinite; therefore, SVD, instead of EVD, was used in [29]. This is the reason why we used SVD in Section 7.2, when implementing FLOM-ESPRIT.




(a) Mean Squared Error ($\alpha = 2.0$).




(b) Mean Squared Error ($\alpha = 1.8$).




(c) Mean Squared Error ($\alpha = 1.5$).




(d) Mean Squared Error ($\alpha = 1.2$).

Figure 7.1: Performance of FLOM-ESPRIT in the presence of $S\alpha S$ noise.

## 7.5 Conclusions

We have proposed a class of FLOM-based matrices that can be used with ESPRIT to estimate the DOAs of independent circular signals embedded in additive $S\alpha S$ noise. Our FLOM-ESPRIT uses $2M$ sensors (two identical subarrays) to estimate the DOAs of up to $M - 1$ signals, and requires less computations than required by FLOM-MUSIC that uses an $M$-element array. The displacement between the two subarrays is less than a half wavelength so as to avoid spatial aliasing. Our FLOM-ESPRIT works if matrix $\Lambda$, defined in (7.7), is non-singular, and the FLOM $p$ satisfies $1 < p < \alpha < 2$. We ran simulations for different choices of FLOMs, and concluded that when the noise is $S\alpha S$ with $\alpha \neq 2$, the FLOM $p$ should be selected close to unity.

# Chapter 8

# Future Research

Since the work in [13], the applications of HOS to a variety of HOS array signal processing problems have been investigated. In this dissertation, we have studied three more problems related to array signal processing in Chapters 2-5. Our studies of these problems show that the advantages (robustness to colored Gaussian noise and hardware savings) of using HOS are preserved. One important work is to test these HOS-based algorithms on real data, because not too much work has been done and reported on this. We leave this to future work.

In some array signal processing scenarios, the additive noise does not have bounded second-order statistics and exhibits impulsive phenomena. $S\alpha S$ processes have been used to characterize such noise. The applications of FLOM to array signal processing scenarios where the noise is $S\alpha S$ have become attractive. In Chapters 6 and 7, we showed that if the signals are circular, then MUSIC and ESPRIT can be applied to the FLOM-based matrices for DOA estimation. Although the signals of interest are limited to be circular, circular signals are fairly common in communication systems. Based on FLOM-MUSIC and FLOM-ESPRIT, we can also apply FLOMs to other applications of array signal processing; one future work is to use beamforming to enhance signal to noise condition so as to improve performance.

Our FLOM-MUSIC and FLOM-ESPRIT are limited to independent circular signals; however, in some communication environments, signals may undergo several multipaths and may no longer be independent. One solution to deal with multipath effects is to use spatial smoothing [70]. Future work is to combine spatial smoothing with either FLOM-MUSIC or FLOM-ESPRIT to combat multipath effects.

# Appendix A

# Proof of Theorem 6.1

Since $C_{ik}$ is a complex number, we show that both its real and imaginary parts are bounded.

To show that the real part of $C_{ik}$, $\Re\{C_{ik}\}$, is bounded, we show that $\Re\{C_{ik}\}$ is both upper and lower bounded. By definition,

$$\begin{aligned}\Re\{C_{ik}\} &\triangleq \Re\{E\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}\} \\ &= E\{\Re\{x_i(t)|x_k(t)|^{p-2}x_k^*(t)\}\} \qquad \text{(A.1)}\end{aligned}$$

The last equality of (A.1) follows from the fact that for any complex random variable $X = X_1 + jX_2$, $\Re\{E\{X\}\} \triangleq \Re\{E\{X_1 + jX_2\}\} = \Re\{E\{X_1\} + jE\{X_2\}\} = E\{X_1\} = E\{\Re\{X\}\}$. Next, we recall the fact that, for any complex number $X$, $\Re\{X\} \triangleq \Re\{X_1 + jX_2\} = X_1 \leq |X_1| \leq \sqrt{X_1^2 + X_2^2} = |X|$, and apply it to (A.1) with $X = x_i(t)|x_k(t)|^{p-2}x_k^*(t)$, to obtain

$$\Re\{C_{ik}\} \leq E\{|x_i(t)|\,|x_k(t)|^{p-2}\,|x_k^*(t)|\} = E\{|x_i(t)|\,|x_k(t)|^{p-1}\} \qquad \text{(A.2)}$$

Substituting (6.1) into $E\{|x_i(t)|\,|x_k(t)|^{p-1}\}$ and using conditional expectation expressions (see Appendix B) for $s_1(t), s_2(t), \cdots, s_P(t), n_i(t), n_k(t)$, we have

$$\begin{aligned}E\{|x_i(t)|\,|x_k(t)|^{p-1}\} &= E\{|\sum_{m=1}^{P} A_{im}s_m(t) + n_i(t)|\,|\sum_{r=1}^{P} A_{kr}s_r(t) + n_k(t)|^{p-1}\} \\ &= E_{s_1}E_{s_2|s_1}E_{s_3|s_1,s_2}\cdots E_{s_P|s_1,s_2,\cdots,s_{P-1}}E_{n_i,n_k|s_1,s_2,\cdots,s_P} \\ &\quad \{|\sum_{m=1}^{P} A_{im}s_m(t) + n_i(t)|\,|\sum_{r=1}^{P} A_{kr}s_r(t) + n_k(t)|^{p-1}\} \qquad \text{(A.3)}\end{aligned}$$

Note that $X_1 \triangleq \sum_{m=1}^{P} A_{im} s_m(t) + n_i(t)$ and $X_2 \triangleq \sum_{r=1}^{P} A_{kr} s_r(t) + n_k(t)$ are jointly $S\alpha S$ if $\sum_{m=1}^{P} A_{im} s_m(t)$ and $\sum_{r=1}^{P} A_{kr} s_r(t)$ are given constants, and, $n_i(t)$ and $n_k(t)$ are jointly $S\alpha S$. In [71], it is shown that if complex $X_1$ and $X_2$ are jointly $S\alpha S$ random variables, and, $p_1 > 0$ and $p_2 > 0$, then

$$E\{|X_1|^{p_1} \, |X_2|^{p_2}\} < \infty \quad \text{if and only if} \quad p_1 + p_2 < \alpha; \tag{A.4}$$

hence, $E_{n_i,n_k|s_1,s_2,\cdots,s_P}\{|\sum_{m=1}^{P} A_{im} s_m(t)+n_i(t)| \, |\sum_{r=1}^{P} A_{kr} s_r(t)+n_k(t)|^{p-1}\}$ is bounded, i.e.,

$$E_{n_i,n_k|s_1,s_2,\cdots,s_P}\{|\sum_{m=1}^{P} A_{im} s_m(t) + n_i(t)| \, |\sum_{r=1}^{P} A_{kr} s_r(t) + n_k(t)|^{p-1}\} < B < \infty \tag{A.5}$$

The bound $B$ in (A.5) is dependent on $s_1(t), s_2(t), \cdots, s_P(t)$. Since our objective is to show that $\Re\{C_{ik}\}$ is bounded, we can select a constant $B = \tilde{B}$ such that (A.5) is satisfied for all possible values of $s_1(t), s_2(t), \cdots, s_P(t)$, i.e., $\tilde{B}$ is independent of $s_1(t), s_2(t), \cdots, s_P(t)$.

Note that (A.5) holds when $1 + (p-1) = p < \alpha$ and $p - 1 > 0$, i.e., $1 < p < \alpha$, which is the constraint given on the right-hand side of (6.3).

Substituting (A.5) into (A.3) and using **[AS2]** to remove the conditional expectations (see Appendix B), we have

$$\begin{aligned} E\{|x_i(t)| \, |x_k(t)|^{p-1}\} &\leq E_{s_1} E_{s_2|s_1} E_{s_3|s_1,s_2} \cdots E_{s_P|s_1,s_2,\cdots,s_{P-1}}\{\tilde{B}\} \\ &= E_{s_1} E_{s_2} \cdots E_{s_P}\{\tilde{B}\} \\ &= \tilde{B} < \infty, \qquad \text{if} \quad 1 < p < \alpha \leq 2 \end{aligned} \tag{A.6}$$

Substituting (A.6) into (A.2) yields

$$Re\{C_{ik}\} < \tilde{B} < \infty, \qquad \text{if} \quad 1 < p < \alpha \leq 2 \tag{A.7}$$

Next, using the facts that for any complex number $X$, $\Re\{X\} \triangleq \Re\{X_1 + jX_2\} = X_1 \geq -|X_1| \geq -\sqrt{X_1^2 + X_2^2} = -|X|$, and, $X = x_i(t)|x_k(t)|^{p-2}x_k^*(t)$ and (A.1), we have

$$\Re\{C_{ik}\} \geq -E\{|x_i(t)|\,|x_k(t)|^{p-2}\,|x_k^*(t)|\} = -E\{|x_i(t)|\,|x_k(t)|^{p-1}\} \tag{A.8}$$

Substituting (A.6) into (A.8), we have $\Re\{C_{ik}\} > -\tilde{B} > -\infty$, $1 < p < \alpha \leq 2$, which, when combined with (A.7), yields

$$-\infty < -\tilde{B} < \Re\{C_{ik}\} < \tilde{B} < \infty, \qquad \text{if} \quad 1 < p < \alpha \leq 2 \tag{A.9}$$

This completes the proof that $\Re\{C_{ik}\}$ is bounded.

The proof that the imaginary part of $C_{ik}$ is bounded is similar to what we did for $\Re\{C_{ik}\}$, and is, therefore, omitted here. □

# Appendix B

# Theorem B.1

Although Theorem B.1 is proved for real random variables, its generalization to the case of complex random variables is straightforward. In this paper, Theorem B.1 is applied to the case of complex random variables, because the complex envelope representation is used for all signals.

**Theorem B.1** *If $G(X_1, X_2, \cdots, X_n)$ is an arbitrary function of jointly distributed random variables, $X_1, X_2, \cdots, X_n$, then*

$$E\{G(X_1, X_2, \cdots, X_n)\} = E_{X_1} E_{X_2|X_1} \cdots E_{X_n|X_1, X_2, \cdots, X_{n-1}} \{G(X_1, X_2, \cdots, X_n)\} \tag{B.1}$$

*where $E_{X_n|X_1, X_2, \cdots, X_{n-1}}\{G(X_1, X_2, \cdots, X_n)\}$ represents the conditional expectation of $G(X_1, X_2, \cdots, X_n)$ given $X_1, X_2, \cdots, X_{n-1}$. Furthermore, if $X_1, X_2, \cdots, X_n$ are independent, then (B.1) becomes*

$$E\{G(X_1, X_2, \cdots, X_n)\} = E_{X_1} E_{X_2} \cdots E_{X_n} \{G(X_1, X_2, \cdots, X_n)\} \tag{B.2}$$

**Proof**: The proof is to expand the RHS of (B.1) and to apply the chain rule [59] to the joint PDF of $X_1, X_2, \cdots, X_n$.

Expanding all the conditional expectations on the RHS of (B.1), we have

$$\begin{aligned} & E_{X_1} E_{X_2|X_1} \cdots E_{X_n|X_1, X_2, \cdots, X_{n-1}} \{G(X_1, X_2, \cdots, X_n)\} \\ = & \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} \cdots \int_{x_n=-\infty}^{\infty} G(X_1, X_2, \cdots, X_n) \\ & \underbrace{f(x_1) f(x_2|x_1) \cdots f(x_n|x_1, x_2, \cdots, x_{n-1})} \, dx_n dx_{n-1} \cdots dx_1 \end{aligned} \tag{B.3}$$

where $f(x_n|x_1, x_2, \cdots, x_{n-1})$ is the conditional PDF of $X_n$ given $x_1, x_2, \cdots, x_{n-1}$. According to the chain rule [59] for a PDF, the underbraced term in (B.3) is equal to the joint PDF of $X_1, X_2, \cdots, X_n$, $f(x_1, x_2, \cdots, x_n)$; hence, the right-hand side of (B.3) equals $E\{G(X_1, X_2, \cdots, X_n)\}$, which proves (B.1).

If $X_1, X_2, \cdots, X_n$ are independent, then the conditional PDFs on the second line of (B.3) are equal to their corresponding marginal PDFs, and (B.3) is written as

$$\begin{aligned} & E_{X_1} E_{X_2|X_1} \cdots E_{X_n|X_1,X_2,\cdots,X_{n-1}} \{G(X_1, X_2, \cdots, X_n)\} \\ = & \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} \cdots \int_{x_n=-\infty}^{\infty} G(X_1, X_2, \cdots, X_n) f(x_n) \cdots f(x_2) f(x_1) dx_n \cdots dx_1 \\ \triangleq & E_{X_1} E_{X_2} \cdots E_{X_n} \{G(X_1, X_2, \cdots, X_n)\} \end{aligned} \tag{B.4}$$

Substituting (B.4) into (B.1), we have (B.2). □

# Appendix C

# Theorem C.1

**Theorem C.1** *If $\tau_{kq}$, $\forall k$ and $q$, are constants, and, $s_k(t)$ and $n_k(t)$, $\forall k$, follow the assumptions in Section 6.1, then*

$$\begin{aligned} D_{ik} &\triangleq E\{n_i(t)|\sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \\ &= \gamma\, \delta_{ik} \end{aligned} \tag{C.1}$$

*and*

$$\begin{aligned} F_{mk} &\triangleq E\{s_m(t)|\sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \\ &= e^{j\tau_{km}}\, \Lambda_{mm} \end{aligned} \tag{C.2}$$

*where*

$$\gamma \triangleq E\{n_k(t)|\textstyle\sum_{q=1}^{P} s_q(t) + n_k(t)|^{p-2}(\sum_{r=1}^{P} s_r(t) + n_k(t))^*\} \tag{C.3}$$

*and*

$$\Lambda_{km} \triangleq \delta_{km}\, E\{s_m(t)|\textstyle\sum_{q=1}^{P} s_q(t) + n_m(t)|^{p-2}(\sum_{r=1}^{P} s_r(t) + n_m(t))^*\} \tag{C.4}$$

**Proof**:

We first show (C.1), and then (C.2).

For $i \neq k$, $n_i(t)$ is independent of all other signals (by **[AS3]**), and the first line of (C.1) becomes

$$\begin{aligned} D_{ik} &= E\{n_i(t)\}\, E\{|\sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \\ &= 0, \qquad \text{if} \quad i \neq k \end{aligned} \tag{C.5}$$

The last equality of (C.5) follows from **[AS3]**, which assumes zero-mean $n_i(t)$. The first line of (C.1) can now be rewritten as

$$D_{ik} = \delta_{ik}\, E\{n_k(t)|\sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} e^{-j\tau_k} s_r(t) + n_k(t))^*\} \tag{C.6}$$

Using conditional expectation expressions on independent $n_k(t)$, $s_1(t)$, $s_2(t), \cdots$, $s_P(t)$ [see (B.2) in Appendix B], we express (C.6) as

$$\begin{aligned} D_{ik} = \delta_{ik}\, & E_{n_k} E_{s_1} E_{s_2} \cdots E_{s_{P-1}} E_{s_P} \\ & \{n_k(t)|\sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \end{aligned} \tag{C.7}$$

Applying Theorem D.1 (see Appendix D) to (C.7) with $s(t) = s_P(t)$ and $\theta = -\tau_{kP}$, we find that

$$\begin{aligned} D_{ik} = \delta_{ik}\, & E_{n_k} E_{s_1} E_{s_2} \cdots E_{s_{P-1}} E_{s_P} \\ & \{n_k(t)|\textstyle\sum_{q=1}^{P-1} e^{-j\tau_{kq}} s_q(t) + s_P(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P-1} e^{-j\tau_{kr}} s_r(t) + s_P(t) + n_k(t))^*\} \end{aligned} \tag{C.8}$$

Note that the coefficient $e^{-j\tau_{kP}}$ of $s_P(t)$ in (C.7) becomes unity in (C.8).

Applying Theorem D.1 to (C.8), with $s(t) = s_p(t)$ and $\theta = -\tau_{kp}$ for $p = 1, 2, \cdots, P-1$, we find that

$$\begin{aligned} D_{ik} = \delta_{ik}\, & E_{n_k} \\ & \left[ E_{s_1} E_{s_2} \cdots E_{s_P} \{n_k(t)|\sum_{p=1}^{P} s_p(t) + n_k(t)|^{p-2} (\sum_{q=1}^{P} s_q(t) + n_k(t))^*\} \right] \end{aligned} \tag{C.9}$$

Note that the term in the brackets of (C.9) is a function only of $n_k(t)$. Since $\{n_k(t)\}_{k=1}^M$ are identically distributed, we can define a constant $\gamma$, independent of $k$, such that

$$\gamma \triangleq D_{11} = D_{22} = \cdots = D_{MM} \tag{C.10}$$

Equations (C.10) and (C.9) together lead to the second line of (C.1).

Next, we show (C.2). From the first line of (C.2) and Theorem B.1 (see Appendix B) for independent $s_1(t), s_2(t), \cdots, s_P(t), n_k(t)$, we have

$$F_{mk} = E_{n_k} E_{s_1} E_{s_2} \cdots E_{s_P} \{s_m(t)| \sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \tag{C.11}$$

Multiplying the RHS of (C.11) by $e^{j\tau_{km}} e^{-j\tau_{km}} = 1$, we find

$$\begin{aligned} F_{mk} &= e^{j\tau_{km}} E_{n_k} E_{s_1} E_{s_2} \cdots E_{s_P} \\ &\quad \{e^{-j\tau_{km}} s_m(t)| \sum_{q=1}^{P} e^{-j\tau_{kq}} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} e^{-j\tau_{kr}} s_r(t) + n_k(t))^*\} \end{aligned} \tag{C.12}$$

Applying Theorem D.1 to (C.12), with $s(t) = s_r(t)$ and $e^{j\theta} = e^{-j\tau_{kr}}$ (see **[AS1]**) for $r = 1, 2, \cdots, P$, we have

$$\begin{aligned} F_{mk} &= e^{j\tau_{km}} E_{n_k} E_{s_1} E_{s_2} \cdots E_{s_P} \{s_m(t)| \sum_{q=1}^{P} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} s_r(t) + n_k(t))^*\} \\ &= e^{j\tau_{km}} \underbrace{E\{s_m(t)| \sum_{q=1}^{P} s_q(t) + n_k(t)|^{p-2} (\sum_{r=1}^{P} s_r(t) + n_k(t))^*\}}_{\Lambda_{mm}} \end{aligned} \tag{C.13}$$

Note that the underbraced term in (C.13) is independent of index $k$, because $n_k(t)$, $\forall k$, are *i.i.d.* Only diagonal entries, $\Lambda_{mm}$, of $\Lambda$ are defined in (C.13); the non-diagonal entries of $\Lambda$ are filled with zeros to make a complete definition of matrix $\Lambda$. We used the Kronecker delta in (C.4). The proof for (C.2) is complete. □

# Appendix D

# Theorem D.1

**Theorem D.1** *If $s(t) = u(t)e^{j\psi(t)}$, where $u(t)$ is a real random process representing the amplitude of $s(t)$, and, $\psi(t)$ is the phase that is uniform over $[0, 2\pi)$ and is independent of $u(t)$, then*

$$E\{g(s(t)e^{j\theta})\} = E\{g(s(t))\} \tag{D.1}$$

*where $g(\cdot)$ is an arbitrary function and $\theta$ is an arbitrary real constant.*

**Proof**: By definition of $s(t)$, and the fact that $\psi(t)$ is uniformly distributed over $[0, 2\pi)$, we see that

$$\begin{aligned} E\{g(s(t)e^{j\theta})\} &= E\{g(u(t)e^{j\psi(t)}e^{j\theta})\} \\ &= \int_0^\infty \int_0^{2\pi} g(u(t)e^{j(\psi(t)+\theta)}) f_{u,\psi}(u(t), \psi(t))\, d\psi(t)\, du(t) \\ &= \int_0^\infty \int_0^{2\pi} g(u(t)e^{j(\psi(t)+\theta)}) \underbrace{f_u(u(t))\frac{1}{2\pi}}_{f_{u,\psi}(u(t),\psi(t))}\, d\psi(t) du(t) \end{aligned} \tag{D.2}$$

where $f_{u,\psi}(u(t), \psi(t))$ is the joint PDF of $u(t)$ and $\psi(t)$, and, $f_u(u(t))$ is the PDF of $u(t)$. The last equality of (D.2) follows from the facts that $u(t)$ is independent of $\psi(t)$, and that the PDF of $\psi(t)$ is $\frac{1}{2\pi}$. By change of variable, $\beta = \psi(t) + \theta$, (D.2) becomes

$$E\{g(s(t)e^{j\theta})\} = \int_0^\infty \left[ \int_\theta^{\theta+2\pi} g(u(t)e^{j\beta}) f_u(u(t)) \frac{1}{2\pi}\, d\beta \right] du(t) \tag{D.3}$$

The integrand in the brackets of (D.3) is a periodic function of $\beta$ with period $2\pi$, and the integration is over a period, starting at $\theta$ and ending at $\theta + 2\pi$. Note that the inner integration of a periodic function over an arbitrary period is constant, no matter which period is selected. Finally, we change the inner integration bounds back to 0 and $2\pi$ and obtain

$$E\{g(s(t)e^{j\theta})\} = \int_0^\infty \int_0^{2\pi} g(u(t)e^{j\beta}) \underbrace{f_u(u(t))\frac{1}{2\pi}}_{f_{u,\beta}(u(t),\beta)} \, d\beta \, du(t) \triangleq E\{g(s(t))\} \qquad \text{(D.4)}$$

# Reference List


[1] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control,* vol. 19, no. 6, pp. 716-723, December 1974.

[2] A. Barabell, "Improving the resolution of eigenstructured based direction finding algorithms," *Proc. ICASSP,* pp. 336-339, 1983.

[3] J. Capon, "High-resolution frequency wavenumber spectral analysis," *Proc. IEEE,* vol. 57, no. 8, pp. 1408-1418, August 1969.

[4] J. A. Cadzow, "A high-resolution direction of arrival algorithm for narrowband coherent and incoherent sources," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 36, pp. 965-977, July 1988.

[5] J.-F. Cardoso and E. Moulines, "Asymptotic performance analysis of direction-finding algorithms based on fourth-order cumulants," *IEEE Trans. Signal Processing,* vol. 43, no. 1, pp. 214-224, January 1995.

[6] R. Challa and S. Shamsunder, "Passive near-field localization of multiple non-Gaussian sources in 3D using cumulants," *Signal Processing,* vol. 65, no. 1, pp. 39-53, February 1998.

[7] H. H. Chiang and C. L. Nikias, "The ESPRIT algorithm with higher-order statistics," *Proceedings of Workshop on Higher-Order Spectral Analysis,* Vail, Colorado, pp. 163-168, June 1989.

[8] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE,* vol. 78, no. 8, pp. 1327-1343, August 1990.

[9] R. D. Degroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing,* vol. 40, no. 3, pp. 571-577, March 1992.

[10] R. D. Degroat, E. M. Dowling, and D. A. Linebarger, "Subspace Tracking," *in Digital Signal Processing Handbook,* IEEE/CRC Press, Ch. 66, December 1997.

[11] J. Dehaene, M. Moonen, and J. Vandewalle, "An improved stochastic gradient algorithm for principal component analysis and subspace tracking," *IEEE Trans. Signal Processing,* vol. 45, no. 10, pp. 2582-2586, October 1997.

[12] M. C. Dogan and J. M. Mendel, "Cumulant-based blind optimum beamforming," *IEEE Trans. Aerospace and Electronic Systems,* vol. 30, pp. 722-741, July 1994.

[13] M. C. Dogan and J. M. Mendel, "Applications of cumulants to array processing, Part I: aperture extension and array calibration," *IEEE Trans. Signal Processing*, vol. 43, no. 5, pp. 1200-1216, May 1995.

[14] M. C. Dogan and J. M. Mendel, "Applications of cumulants to array processing, Part II: non-Gaussian noise suppression," *IEEE Trans. Signal Processing*, vol. 43, pp. 1663-1676, July 1995.

[15] E. M. Dowling and R. D. Degroat, "Adaptation dynamics of the spherical subspace tracker," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2599-2602, October 1992.

[16] E. M. Dowling, L. P. Ammaann, and R. D. Degroat, "A TQR-iteration based adaptive SVD for real time angle and frequency tracking," *IEEE Trans. Signal Processing*, vol. 42, no. 4, pp. 914-926, April 1994.

[17] A. Eriksson, P. Stoica, and T. Soderstrom, "On-line subspace algorithms for tracking moving source," *IEEE Trans. Signal Processing*, vol. 42, no. 9, pp. 2319-2330, September 1994.

[18] B. Friedlander, "A sensitivity analysis of the MUSIC algorithm," *IEEE Trans. Signal Processing*, vol. 38, no. 10, pp. 1740-1751, October 1990.

[19] Z. Fu and E. M. Dowling, "Conjugate gradient projection subspace tracking," *IEEE Trans. Signal Processing*, vol. 45, no. 6, pp. 1664-1668, June 1997.

[20] G. Giannakis and J. M. Mendel, "Identification of nonminimum phase systems using higher-order statistics," *IEEE Trans. Acoustic, Speech, Signal Processing*, vol. 37, no. 3, pp. 360-377, March 1989.

[21] L. C. Godara, "Application of antenna arrays to mobile communications, Part I: performance improvement, feasibility, and system considerations," *Proc. IEEE,* vol. 85, no. 7, pp. 1031-1060, July 1997.

[22] L. C. Godara, "Application of antenna arrays to mobile communications, Part II: beamforming and direction of arrival considerations," *Proc. IEEE,* vol. 85, no. 8, pp. 1195-1245, August 1997.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, Maryland, University Press, 1983.

[24] G. H. Golub, *Matrix Computations. 3rd Ed.,* Johns Hopkins University Press, 1996.

[25] E. Gonen and J. M. Mendel, "Beamspace virtual-ESPRIT," *29th Asilomar Conf. Signals, Systems, and Computers,* November 1995.

[26] E. Gonen and J. M. Mendel, "An iterative virtual-ESPRIT algorithm," *8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing,* pp. 86-89, June 1996, Corfu, Greece.

[27] E. Gonen, *Cumulant and Subspace Techniques for Array Processing.* Ph.D Dissertation, Signal and Image Processing Institute Report #309, University of Southern California, May 1997.

[28] E. Gonen and J. M. Mendel, "Applications of cumulants to array processing, Part III: blind beamforming for coherent signals," *IEEE Trans. Signal Processing,* vol. 45, no. 9, pp. 2252-2264, September 1997.

[29] E. Gonen, J. M. Mendel, and M. C. Dogan, "Applications of cumulants to array processing, Part IV: direction finding in coherent signals case," *IEEE Trans. Signal Processing,* vol. 45, no. 9, pp. 2265-2276, September 1997.

[30] E. Gonen and J. M. Mendel, "Subspace-based direction finding," *in Digital Signal Processing Handbook,* IEEE/CRC Press, Ch. 62, December 1997.

[31] R. Hamza and K. Buckley, "Resolution enhanced ESPRIT," *IEEE Trans. Signal Processing,* vol. 42, no. 3, pp. 688-691, March 1994.

[32] R. Hamza and K. Buckley, "An analysis of weighted eigenspace methods in the presence of sensor errors," *IEEE Trans. Signal Processing,* vol. 43, no. 5, pp. 1140-1150, May 1995.

[33] D. Hatzinakos, C. L. Nikias, "Blind equalization using a tricepstrum-based algorithms," *IEEE Trans. Communications,* vol. 39, no. 5, pp. 669-682, May 1991.

[34] S. Haykin (ed.), *Array Signal Processing.* Prentice Hall, 1985.

[35] S. Haykin, *Adaptive Filter Theory.* 2nd Ed., Prentice Hall, 1991.

[36] S. Haykin, J. Litva, and T. J. Shepherd (eds.), *Radar Array Processing.* Springer-Verlag, 1993.

[37] A. Janicki and A. Weron, *Simulation and Chaotic Behaivior of $\alpha$-stable Stochastic Processes,* Dekker, New York, 1993.

[38] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques.* Prentice Hall, 1993.

[39] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Signal Processing*, vol. 34, no. 2, pp. 8-12, February 1986.

[40] A. Kavcic and B. Yang, "Adaptive rank estimation for spherical subspace trackers," *IEEE Trans. Signal Processing*, vol. 44, no. 6, pp. 1573-1579, June 1996.

[41] H. Krim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine,* vol. 13, no. 4, pp. 67-94, July 1996.

[42] S. M. Kuo and D. R. Morgan, *Active noise control systems: algorithms and DSP implementation.* John Wiley and Sons, 1996

[43] S. S. Kuzin and M. V. Ratynskii, "Superresolution direction finding by the method of reflection of subspaces," *Journal of Communications Technology and Electronics,* vol. 41, no. 12, pp. 1054-1057, 1996.

[44] H. B. Lee and M. S. Wengrovitz, "Resolution threshold of beamspace MUSIC for two closely spaced emitters," *IEEE Trans. Acoustic, Speech, and Signal Processing,* vol. 38, no. 9, pp. 1545-1559, September 1990.

[45] F. Li and R. J. Vaccaro, "Performance degradation of DOA estimators due to unknown noise fields," *IEEE Trans. Signal Processing,* vol. 40, no. 3, pp. 686-689, March 1992.

[46] J. Li and R. T. Compton, Jr., "Two-dimensional angle and polarization estimation using the ESPRIT algorithm," *IEEE Trans. Antennas and Prop.*, vol. 40, no. 5, pp. 550-555, May 1992.

[47] F. Li and R. J. Vaccaro, "Sensitivity analysis of DOA estimation algorithms to sensor errors," *IEEE Trans. Aerospace and Electronic Systems,*

[48] D. A. Linebarger, R. D. Degroat, and E. M. Dowling, "Efficient direction-finding methods employing forward/backward averaging," *IEEE Trans. Signal Processing*, vol. 42, no. 8, pp. 2136-2145, August 1994.

[49] K. J. Liu, D. P. O'Leary, G. R. Stewart, and Y. J. Wu, "URV ESPRIT for tracking time-varying signals," *IEEE Trans. Signal Processing*, vol. 42, no. 12, pp. 3441-3448, December 1994.

[50] T.-H. Liu and J. M. Mendel, "The robustness of virtual-ESPRIT against model errors," *31th Asilomar Conf. Signals, Systems, and Computers,* November 1997.

[51] X. Ma and C. L. Nikias, "Joint estimation of time delay and frequency delay in impulsive noise," *IEEE Trans. Signal Processing,* vol. 44, no. 11, pp. 2669-2687, November 1996.

[52] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications," *Proc. IEEE,* vol. 79, pp. 278-305, March 1991.

[53] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control.* Prentice Hall, 1995.

[54] E. K. Miller, "Model-based parameter estimation applications in electromagnetics," *Electromagnetic Modeling and Measurements for Analysis and Synthesis Problems.* Bernard de Neumann, Ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 205-256, 1991.

[55] B. D. Moor, "The singular value decomposition and long and short spaces of noisy matrices," *IEEE Trans. Signal Processing,* vol. 41, no. 9, pp. 2826-2838, September 1993.

[56] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis - A Nonlinear Signal Processing Framework.* Prentice Hall, 1993.

[57] S. J. Orfanidis, *Optimum Signal Processing: An Introduction.* Macmillan, 1988.

[58] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Processing,* vol. 40, pp. 590-600, March 1992.

[59] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* McGraw Hill, 1991.

[60] R. L. Peterson, R. E. Ziemer, and D.E. Borth, *Introduction to Spread Spectrum Communications.* Prentice Hall, 1995.

[61] S. U. Pillai, *Array Signal Processing.* Springer-Verlag, 1989.

[62] H. V. Poor and G. W. Wornell, *Wireless Communications: Signal Processing Perspectives.* Prentice Hall, 1998.

[63] B. Porat and B. Friedlander, "Direction finding algorithms based on high-order statistics," *IEEE Trans. Signal Processing,* vol. 39, no. 9, pp. 2016-2023, September 1991.

[64] D. J. Rabideau, "Subspace invariance: the RO-FST and TQR-SVD adaptive subspace tracking algorithms," *IEEE Trans. Signal Processing,* vol. 44, no. 8, pp. 2016-2018, August 1995.

[65] D. J. Rabideau, "Fast, rank adaptive subspace tracking and applications," *IEEE Trans. Signal Processing,* vol. 44, no. 9, pp. 2229-2244, September 1996.

[66] B. D. Rao and K. V. Hari, "Performance analysis of ESPRIT and TAM in determining the directions of arrival of plane waves in noise," *IEEE Trans. Signal Processing*, vol. 37, no. 12, pp. 1990-1995, December 1989.

[67] R. Roy and T. Kailath, "ESPRIT - Estimation of signal parameters via rotational invariance techniques," *Optical Engineering*, vol.29, no.4, pp. 296-313, April 1990.

[68] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antenna and Propagation,* vol. 34, no. 3, pp. 276-280, March 1986.

[69] G. Shamorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance.* Chapman Hall, New York, 1994.

[70] T. J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoustic Speech, Signal Processing*, vol. 33, no. 4, pp. 806-811, August 1985.

[71] M. Shao and C. L. Nikias, *Signal Processing With Alpha-Stable Distributions and Applications.* Wiley, New York, 1995.

[72] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, no. 6, pp. 1535-1541, June 1992.

[73] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Trans. Acoustic Speech, Signal Processing*, vol. 37, no. 5, pp. 720-741, May 1989.

[74] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Trans. Acoustic Speech, Signal Processing*, vol. 38, no. 10, pp. 1783-1795, October 1990.

[75] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. Signal Processing*, vol. 45, no. 5, pp. 1222-1240, May 1997.

[76] A. Swami and B. Sadler, "TDE, DOA and related parameter estimation problems in impulsive noise," *Proc. IEEE Signal Processing Workshops on Higher-Order Statistics,* pp. 273-277, Banff, Canada, July 1997.

[77] A. L. Swindlehurst, B. Ottersten, R. Roy, and T. Kailath, "Multiple invariance ESPRIT," *IEEE Trans. Signal Processing,* vol. 40, no. 4, pp. 867-881, April 1992.

[78] A. L. Swindlehurst and T. Kailath, "A performance analysis of subspace-based methods in the presence of model errors, Part I: the MUSIC algorithm," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1758-1773, July 1992.

[79] A. Swindlehurst and T. Kailath, "Azimuth/elevation direction finding using regular array geometries," *IEEE Trans. Aero. Electron. Syst.*, vol. 29, no. 1, pp. 145-156, January 1993.

[80] A. L. Swindlehurst and T. Kailath, "A performance analysis of subspace-based methods in the presence of model errors, Part II: Multidimensional algorithms," *IEEE Trans. Signal Processing*, vol. 41, no. 9, pp. 1277-1308, September 1993.

[81] P. A. Thompson, "An adaptive spectral analysis technique for unbiased frequency estimation in the presence of white noise," *Proceedings of 13th Asilomar Conference on Circuits, System, and Computers,* Pacific Grove, California, 1980.

[82] G. A. Tsihrintzis and C. L. Nikias, "Fast estimation of the parameters of alpha-stable impulsive interference," *IEEE Trans. Signal Processing,* vol. 44, no. 6, pp. 1492-1503, June 1996.

[83] P. Tsakalides and C. L. Nikias, "Maximum likelihood localization of sources in noise modeled as a stable process," *IEEE Trans. Signal Processing,* vol. 43, no. 11, pp. 2700-2713, November 1995.

[84] P. Tsakalides and C. L. Nikias, "Wideband array signal processing with alpha-stable distributions," *Proc. IEEE Military Communications Conf.,* November 1995.

[85] P. Tsakalides and C. L. Nikias, "Broadband beamforming with alpha-stable distribution," *29th Asilomar Conference on Signals, Systems and Computers,* November 1995.

[86] P. Tsakalides, R. Raspanti, and C. L. Nikias, "Joint target angle and doppler estimation in interference modeled as a stable process," *30th Conf. Information Sciences and Systems,* Princeton, New Jersey, March 1996.

[87] P. Tsakalides and C. L. Nikias, "Robust adaptive beamforming in alpha-stable noise environments," *IEEE ICASSP-96,* vol. 5, pp. 2884-2487, May 1996.

[88] P. Tsakalides and C. L. Nikias, "The robust covariation-based MUSIC (ROC-MUSIC) algorithm for bearing estimation in impulsive noise environments," *IEEE Trans. Signal Processing,* vol. 44, no. 7, pp. 1623-1633, July 1996.

[89] R. J. Vaccaro, "The past, present, and future of underwater acoustic signal processing," *IEEE Signal Processing Magazines,* vol. 15, no. 4, pp. 21-53, July, 1998.

[90] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE Signal Processing Magazine,* vol. 5, no. 2, pp. 4-24, February 1988.

[91] A. Veen, P. Ober, and E. Deprettere, "Azimuth and elevation computation in high resolution DOA estimation," *IEEE Trans. Signal Processing,* vol. 40, no. 7, pp. 1828-1832, April 1985.

[92] M. Viberg and L. Swindlehurst, "Analysis of combined effects of finite samples and model errors on array processing performance," *IEEE Trans. Signal Processing,* vol. 42, no. 11, pp. 3073-3083, November 1994.

[93] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 33, no. 2, pp. 387-392, April, 1985.

[94] A. J. Weiss and B. Friedlander, "Array processing using joint diagonalization," *Signal Processing,* vol. 50, no. 3, pp. 205-222, May 1996.

[95] G. Xu, S. D. Silverstein, R. H. Roy, and T. Kailath, "Beamspace ESPRIT," *IEEE Trans. Signal Processing,* vol. 42, no. 2, pp. 349-356, February 1994.

[96] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing,* vol. 43, no. 1, pp. 95-107, January 1995.

[97] J.-F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 35, no. 2, pp. 241-251, February 1988.

[98] J.-F. Yang, H.-T. Wu, and F.-K. Chen, "Simplified adaptive noise subspace algorithms for robust direction tracking," *IEE Proc. Radar and Signal Processing,* vol. 140, pp. 329-34, October 1993.

[99] Q. Yin, R. Newcomb, and L. Zou, "Estimating 2-D angles of arrival via two parallel linear arrays," *Proc. ICASSP,* pp. 2803-2806, Glasgow, Scotland, May 1989.

[100] K.-B. Yu, "Recursive updating the eigenvalue decomposition of a covariance matrix," *IEEE Trans. Signal Processing,* vol. 39, no. 5, pp. 1136-1145, May 1991.

[101] N. Yuen and B. Friedlander, "Asymptotic performance analysis of ESPRIT, higher order ESPRIT, and Virtual ESPRIT algorithms," *IEEE Trans. Signal Processing,* vol. 44, no.10, pp. 2537-2550, October 1996

[102] M. D. Zoltowski and D. Stavrinides, "Sensor array signal processing via a procrustes rotations based eigenanalysis of the ESPRIT data pencil," *IEEE Trans. Acoustic, Speech, and Signal Processing,* vol. 37, no. 6, pp. 2537-2550, June 1989.

[103] M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beamspace root-MUSIC," *IEEE Trans. Signal Processing*, vol. 41, no. 1, pp. 344-346, January 1993.